

[illegible]

[illegible]



```
1 0001 0 MODULE JBCCMDPRS(XTITLE 'Job Controller Command Parsing Utilities'
2 0002 0 IDENT = 'V04-000'
3 0003 0 ) =
4 0004 1 BEGIN
5 0005 1
6 0006 1
7 0007 1 *****
8 0008 1 *
9 0009 1 * COPYRIGHT (c) 1978, 1980, 1982, 1984 BY
10 0010 1 * DIGITAL EQUIPMENT CORPORATION, MAYNARD, MASSACHUSETTS.
11 0011 1 * ALL RIGHTS RESERVED.
12 0012 1 *
13 0013 1 * THIS SOFTWARE IS FURNISHED UNDER A LICENSE AND MAY BE USED AND COPIED
14 0014 1 * ONLY IN ACCORDANCE WITH THE TERMS OF SUCH LICENSE AND WITH THE
15 0015 1 * INCLUSION OF THE ABOVE COPYRIGHT NOTICE. THIS SOFTWARE OR ANY OTHER
16 0016 1 * COPIES THEREOF MAY NOT BE PROVIDED OR OTHERWISE MADE AVAILABLE TO ANY
17 0017 1 * OTHER PERSON. NO TITLE TO AND OWNERSHIP OF THE SOFTWARE IS HEREBY
18 0018 1 * TRANSFERRED.
19 0019 1 *
20 0020 1 * THE INFORMATION IN THIS SOFTWARE IS SUBJECT TO CHANGE WITHOUT NOTICE
21 0021 1 * AND SHOULD NOT BE CONSTRUED AS A COMMITMENT BY DIGITAL EQUIPMENT
22 0022 1 * CORPORATION.
23 0023 1 *
24 0024 1 * DIGITAL ASSUMES NO RESPONSIBILITY FOR THE USE OR RELIABILITY OF ITS
25 0025 1 * SOFTWARE ON EQUIPMENT WHICH IS NOT SUPPLIED BY DIGITAL.
26 0026 1 *
27 0027 1 *
28 0028 1 *****
29 0029 1
30 0030 1
31 0031 1 ++
32 0032 1 FACILITY:
33 0033 1 Queue manipulation commands.
34 0034 1
35 0035 1 ABSTRACT:
36 0036 1 This module contains common qualifier parsing routines for the queue
37 0037 1 manipulation commands.
38 0038 1
39 0039 1 ENVIRONMENT:
40 0040 1 VAX/VMS user mode.
41 0041 1 --
42 0042 1
43 0043 1 AUTHOR: M. Jack, CREATION DATE: 30-Apr-1982
44 0044 1
45 0045 1 MODIFIED BY:
46 0046 1
47 0047 1 V03-101 JAK0220 J A Krycka 20-Jul-1984
48 0048 1 Add support for new qualifiers.
49 0049 1
50 0050 1 V03-009 MLJ0118 Martin L. Jack, 23-Aug-1983
51 0051 1 Change names to track $SJCDEF.
52 0052 1
53 0053 1 V03-007 LMP0140 L. Mark Pilant, 23-Aug-1983
54 0054 1 Add support for alphanumeric UICs.
55 0055 1
56 0056 1 V03-007 MLJ0115 Martin L. Jack, 30-Jul-1983
57 0057 1 Changes for job controller baselevel.
```



58	0058	1	:			
59	0059	1	:	V03-006	KBT0569	Keith B. Thompson
60	0060	1	:		Remove lib\$extract_conceal hack	29-Jul-1983
61	0061	1	:			
62	0062	1	:	V03-005	MLJ0114	Martin L. Jack, 23-Jun-1983
63	0063	1	:		Changes for job controller baselevel.	
64	0064	1	:			
65	0065	1	:	V03-004	MLJ0113	Martin L. Jack, 26-May-1983
66	0066	1	:		Changes for job controller baselevel.	
67	0067	1	:			
68	0068	1	:	V03-003	MLJ0112	Martin L. Jack, 29-Apr-1983
69	0069	1	:		Changes for job controller baselevel.	
70	0070	1	:			
71	0071	1	:	V03-002	MLJ0109	Martin L. Jack, 13-Apr-1983
72	0072	1	:		Add new qualifiers, relax length restrictions, disallow wildcard	
73	0073	1	:		in /LOG_FILE.	
74	0074	1	:			
75	0075	1	:	V03-001	MLJ0106	Martin L. Jack, 01-Mar-1983
76	0076	1	:		Support START/QUEUE/MANAGER.	
77	0077	1	:			
78	0078	1	:			



```
80 0079 1 LIBRARY 'SYSSLIBRARY:LIB';
81 0080 1 LIBRARY 'SYSSLIBRARY:TPAMAC';
82 0081 1 REQUIRE 'SRC$:JBCPRSDEF';
83 0191 1
84 0192 1
85 0193 1 LITERAL
86 0194 1 TRUE= 1;
87 0195 1 FALSE= 0;
88 0196 1
89 0197 1
90 0198 1 STRUCTURE
91 0199 1 BBLOCK[O,P,S,E;N]=
92 0200 1 [N]
93 0201 1 (BBLOCK + 0)<P,S,E>;
94 0202 1
95 0203 1
96 0204 1 PSECT
97 0205 1 CODE= CODE,
98 0206 1 PLIT= CODE,
99 0207 1 OWN= DATA,
100 0208 1 GLOBAL= DATA;
101 0209 1
102 0210 1
103 0211 1 FORWARD ROUTINE
104 0212 1 CALL TPARSE,
105 0213 1 PARSE_IF_TRUE: PARSE_LINKAGE,
106 0214 1 PARSE_IF_FALSE: PARSE_LINKAGE,
107 0215 1 PARSE_IF_TRUE_FALSE: PARSE_LINKAGE,
108 0216 1 PARSE_LOCAL_TRUE_FALSE: PARSE_LINKAGE,
109 0217 1 PARSE_AFTER: PARSE_LINKAGE,
110 0218 1 PARSE_ALIGN: PARSE_LINKAGE,
111 0219 1 PARSE_BACKWARD: PARSE_LINKAGE,
112 0220 1 PARSE_BASE_PRIORITY: PARSE_LINKAGE,
113 0221 1 PARSE_BUFFER_COUNT: PARSE_LINKAGE,
114 0222 1 PARSE_CHARACTERISTICS: PARSE_LINKAGE,
115 0223 1 PARSE_CHAR_NUMBER: PARSE_LINKAGE,
116 0224 1 PARSE_COMMA_LIST: PARSE_LINKAGE,
117 0225 1 PARSE_COPIES: PARSE_LINKAGE,
118 0226 1 PARSE_CPUTIME: PARSE_LINKAGE,
119 0227 1 PARSE_ENTRY: PARSE_LINKAGE VALUE,
120 0228 1 PARSE_EXTEND QUANTITY: PARSE_LINKAGE,
121 0229 1 PARSE_FILENAME: PARSE_LINKAGE,
122 0230 1 PARSE_FORWARD: PARSE_LINKAGE,
123 0231 1 PARSE_FORM: PARSE_LINKAGE,
124 0232 1 PARSE_GENERIC: PARSE_LINKAGE,
125 0233 1 PARSE_JOB_LIMIT: PARSE_LINKAGE,
126 0234 1 PARSE_LOG_FILE: PARSE_LINKAGE VALUE,
127 0235 1 PARSE_LOWER_UPPER: PARSE_LINKAGE,
128 0236 1 PARSE_NAME: PARSE_LINKAGE,
129 0237 1 PARSE_NAME AND LOG FILE: PARSE_LINKAGE,
130 0238 1 PARSE_NONZERO_NUMBER: PARSE_LINKAGE,
131 0239 1 PARSE_NUMBER: PARSE_LINKAGE,
132 0240 1 PARSE_OBJECT_NAME: PARSE_LINKAGE,
133 0241 1 PARSE_ON: PARSE_LINKAGE,
134 0242 1 PARSE_OWNER: PARSE_LINKAGE,
135 0243 1 PARSE_PARAMETERS: PARSE_LINKAGE,
136 0244 1 PARSE_PRINTER: PARSE_LINKAGE,
```



```
: 137      0245 1      PARSE_PRIORITY:      PARSE_LINKAGE,
: 138      0246 1      PARSE_PROTECTION:    PARSE_LINKAGE,
: 139      0247 1      PARSE_QUEUE:          PARSE_LINKAGE,
: 140      0248 1      PARSE_SEARCH_STRING:  PARSE_LINKAGE,
: 141      0249 1      PARSE_STRING:         PARSE_LINKAGE,
: 142      0250 1      PARSE_STRING_255:     PARSE_LINKAGE,
: 143      0251 1      PARSE_WORKING_SET:    PARSE_LINKAGE;
: 144      0252 1
: 145      0253 1
: 146      0254 1  EXTERNAL ROUTINE
: 147      0255 1      CLISGET_VALUE: ADDRESSING_MODE(GENERAL),
: 148      0256 1                      ! Get a parameter or qualifier value
: 149      0257 1      CLISPRESENT: ADDRESSING_MODE(GENERAL),
: 150      0258 1                      ! Determine if entity is present
: 151      0259 1      LIB$CVT_DTB: ADDRESSING_MODE(GENERAL),
: 152      0260 1                      ! Convert decimal string to binary
: 153      0261 1      LIB$CVT_DTIME: ADDRESSING_MODE(GENERAL),
: 154      0262 1                      ! Convert delta time to binary
: 155      0263 1      LIB$CVT_TIME: ADDRESSING_MODE(GENERAL),
: 156      0264 1                      ! Convert absolute time to binary
: 157      0265 1      LIB$TPARSE: ADDRESSING_MODE(GENERAL);
: 158      0266 1                      ! Table-driven parser
: 159      0267 1
: 160      0268 1
: 161      0269 1  EXTERNAL LITERAL
: 162      0270 1      CLIS_COMMA,
: 163      0271 1      CLIS_NEGATED,
: 164      0272 1      CLIS_LOCNeg,
: 165      0273 1      CLIS_LOCPRES;
: 166      0274 1
: 167      0275 1
: 168      0276 1  BIND
: 169      0277 1      LITERAL_MINUS_ONE=      UPLIT(-1),
: 170      0278 1      LITERAL_ZERO=          UPLIT(0),
: 171      0279 1      LITERAL_ONE=           UPLIT(+1);
: 172      0280 1
: 173      0281 1
: 174      0282 1  FORWARD
: 175      0283 1
: 176      0284 1      ! TPARSE tables, which are defined at the end of this module.
: 177      0285 1      !
: 178      0286 1      NONE_STATES: VECTOR[0],
: 179      0287 1      NONE_KEYS: VECTOR[0],
: 180      0288 1      INFI_STATES: VECTOR[0],
: 181      0289 1      INFI_KEYS: VECTOR[0],
: 182      0290 1      SYMB_STATES: VECTOR[0],
: 183      0291 1      SYMB_KEYS: VECTOR[0],
: 184      0292 1      MASK_STATES: VECTOR[0],
: 185      0293 1      MASK_KEYS: VECTOR[0],
: 186      0294 1      OWNE_STATES: VECTOR[0],
: 187      0295 1      OWNE_KEYS: VECTOR[0],
: 188      0296 1      PROT_STATES: VECTOR[0],
: 189      0297 1      PROT_KEYS: VECTOR[0];
: 190      0298 1
: 191      0299 1
: 192      0300 1  OWN
: 193      0301 1      TPA_1,
```



JBCCMDPRS  
V04-000

Job Controller Command Parsing Utilities

E 1  
16-Sep-1984 00:09:18  
14-Sep-1984 12:08:34

VAX-11 Bliss-32 V4.0-742  
[CLIUTL.SRC]JBCCMDPRS.B32;1

Page 5  
(2)

:	194	0302	1		CONVERTED_UIC;
:	195	0303	1		
:	196	0304	1		
:	197	0305	1	BUILTIN	
:	198	0306	1		EDIV,
:	199	0307	1		TESTBITCC;



```
201 0308 1 ROUTINE CALL_TPARSE(SRC,STATES,KEYS)=
202 0309 1
203 0310 1 ++
204 0311 1
205 0312 1 FUNCTIONAL DESCRIPTION:
206 0313 1 This routine executes a call to LIB$TPARSE.
207 0314 1
208 0315 1 INPUT PARAMETERS:
209 0316 1 SRC - Address of a descriptor for the string to be parsed.
210 0317 1 STATES - State table parameter for TPARSE.
211 0318 1 KEYS - Keyword table parameter for TPARSE.
212 0319 1
213 0320 1 IMPLICIT INPUTS:
214 0321 1 NONE
215 0322 1
216 0323 1 OUTPUT PARAMETERS:
217 0324 1 NONE
218 0325 1
219 0326 1 IMPLICIT OUTPUTS:
220 0327 1 NONE
221 0328 1
222 0329 1 ROUTINE VALUE:
223 0330 1 As returned by LIB$TPARSE.
224 0331 1
225 0332 1 SIDE EFFECTS:
226 0333 1 NONE
227 0334 1
228 0335 1 --
229 0336 1
230 0337 2 BEGIN
231 0338 2 MAP
232 0339 2 SRC: REF BBLOCK; ! Pointer to descriptor
233 0340 2 LOCAL
234 0341 2 TPA_PARAM: BBLOCK[TPASK_LENGTH0]; ! TPARSE parameter block
235 0342 2
236 0343 2
237 0344 2 CH$FILL(0, TPASK_LENGTH0, TPA_PARAM);
238 0345 2 TPA_PARAM[TPASL_COUNT] = TPASK_COUNT0;
239 0346 2 TPA_PARAM[TPASL_OPTIONS] = TPASM_ABBREV;
240 0347 2 TPA_PARAM[TPASL_STRINGCNT] = .SRC[DSC$W_LENGTH];
241 0348 2 TPA_PARAM[TPASL_STRINGPTR] = .SRC[DSC$A_POINTER];
242 0349 2 LIB$TPARSE(TPA_PARAM, .STATES, .KEYS)
243 0350 1 END;
```

```
.TITLE JBCCMDPRS Job Controller Command Parsing Utilit
ies
```

```
.IDENT \V04-000\
```

```
.PSECT DATA,NOEXE,2
```

```
00000 TPA 1: .BLKB 4
```

```
00004 CONVERTED UIC: .BLKB 4
```

```
.PSECT CODE,NOWRT,2
```



FFFFFFFF 00000 P.AAA: .LONG -1  
00000000 00004 P.AAB: .LONG 0  
00000001 00008 P.AAC: .LONG 1

LITERAL\_MINUS\_ONE= P.AAA  
LITERAL\_ZERO= P.AAB  
LITERAL\_ONE= P.AAC  
.EXTRN CLISGET\_VALUE, CLISPRESNT  
.EXTRN LIB\$CVT\_DTB, LIB\$CVT\_DTIME  
.EXTRN LIB\$CVT\_TIME, LIB\$TPARSE  
.EXTRN CLIS\_COMMA, CLIS\_NEGATED  
.EXTRN CLIS\_LOCNEG, CLIS\_LOCPRES

24	00	5E	24	C2	00002	.WORD	Save R2,R3,R4,R5	: 0308
		6E	00	2C	00005	SUBL2	#36, SP	: 0344
			6E		0000A	MOVCS	#0, (SP), #0, #36, TPA_PARAM	: 0345
		6E	08	D0	0000B	MOVL	#8, TPA_PARAM	: 0346
	04	AE	02	D0	0000E	MOVL	#2, TPA_PARAM+4	: 0347
		50	04	AC	D0 00012	MOVL	SRC, R0	: 0348
	08	AE	60	3C	00016	MOVZWL	(R0), TPA_PARAM+8	: 0349
	0C	AE	04	A0	D0 0001A	MOVL	4(R0), TPA_PARAM+12	: 0350
		7E	08	AC	7D 0001F	MOVQ	STATES, -(SP)	
			08	AE	9F 00023	PUSHAB	TPA_PARAM	
	00000000G	00	03	FB	00026	CALLS	#3, LIB\$TPARSE	
			04	00	0002D	RET		

; Routine Size: 46 bytes, Routine Base: CODE + 000C



```
: 245 0351 1 GLOBAL ROUTINE PARSE_IF_TRUE(PARSE_PARAMETERS_): PARSE_LINKAGE=
: 246 0352 1
: 247 0353 1 !++
: 248 0354 1
: 249 0355 1 FUNCTIONAL DESCRIPTION:
: 250 0356 1 This routine parses a Boolean qualifier, making an entry in the job
: 251 0357 1 controller parameter list if the qualifier is present.
: 252 0358 1
: 253 0359 1 INPUT PARAMETERS:
: 254 0360 1 Standard parser parameters.
: 255 0361 1
: 256 0362 1 IMPLICIT INPUTS:
: 257 0363 1 NONE
: 258 0364 1
: 259 0365 1 OUTPUT PARAMETERS:
: 260 0366 1 NONE
: 261 0367 1
: 262 0368 1 IMPLICIT OUTPUTS:
: 263 0369 1 NONE
: 264 0370 1
: 265 0371 1 ROUTINE VALUE:
: 266 0372 1 NONE
: 267 0373 1
: 268 0374 1 SIDE EFFECTS:
: 269 0375 1 NONE
: 270 0376 1
: 271 0377 1 !--
: 272 0378 1
: 273 0379 2 BEGIN
: 274 0380 2 PARSE_EXTERNAL_REGISTERS; ! Declare external registers
: 275 0381 2
: 276 0382 2
: 277 0383 2 IF CLISPRESNT(.Q_DESC)
: 278 0384 2 THEN
: 279 0385 3 BEGIN
: 280 0386 3 Q_ICURSOR[0,0,16,0] = 0;
: 281 0387 3 Q_ICURSOR[2,0,16,0] = .Q_P1;
: 282 0388 3 Q_ICURSOR[4,0,32,0] = 0;
: 283 0389 3 Q_ICURSOR[8,0,32,0] = 0;
: 284 0390 3 Q_ICURSOR = .Q_ICURSOR + 12;
: 285 0391 2 END;
: 286 0392 1 END;
```

			0000 00000	.ENTRY	PARSE_IF_TRUE, Save nothing	: 0351
		04	AC DD 00002	PUSHL	Q_DESC	: 0383
00000000G	00		01 FB 00005	CALLS	#T, CLISPRESNT	:
	08		50 E9 0000C	BLBC	RO, 1\$	:
			8B B4 0000F	CLRW	(Q_ICURSOR)+	: 0386
	8B	08	AC B0 00011	MOVW	Q_P1, (Q_ICURSOR)+	: 0387
			8B 7C 00015	CLRQ	(Q_ICURSOR)+	: 0388
			04 00017 1\$:	RET		: 0392

; Routine Size: 24 bytes, Routine Base: CODE + 003A



JBCCMDPRS  
V04-000

Job Controller Command Parsing Utilities

<sup>1</sup><sub>1</sub>  
16-Sep-1984 00:09:18  
14-Sep-1984 12:08:34

VAX-11 Bliss-32 V4.0-742  
[CLIUTL.SRC]JBCCMDPRS.B32;1

Page 9  
(4)



```
288 0393 1 GLOBAL ROUTINE PARSE_IF_FALSE(PARSE_PARAMETERS_): PARSE_LINKAGE=
289 0394 1
290 0395 1 !++
291 0396 1
292 0397 1 FUNCTIONAL DESCRIPTION:
293 0398 1 This routine parses a Boolean qualifier, making an entry in the job
294 0399 1 controller parameter list if the qualifier is explicitly negated.
295 0400 1
296 0401 1 INPUT PARAMETERS:
297 0402 1 Standard parser parameters.
298 0403 1
299 0404 1 IMPLICIT INPUTS:
300 0405 1 NONE
301 0406 1
302 0407 1 OUTPUT PARAMETERS:
303 0408 1 NONE
304 0409 1
305 0410 1 IMPLICIT OUTPUTS:
306 0411 1 NONE
307 0412 1
308 0413 1 ROUTINE VALUE:
309 0414 1 NONE
310 0415 1
311 0416 1 SIDE EFFECTS:
312 0417 1 NONE
313 0418 1
314 0419 1 !--
315 0420 1
316 0421 2 BEGIN
317 0422 2 PARSE_EXTERNAL_REGISTERS; ! Declare external registers
318 0423 2 LOCAL
319 0424 2 STATUS;
320 0425 2 BUILTIN
321 0426 2 ACTUALCOUNT,
322 0427 2 ACTUALPARAMETER;
323 0428 2
324 0429 2
325 0430 2 STATUS = CLISPRESNT(.Q DESC);
326 0431 2 IF .STATUS EQL CLIS_NEGATED OR .STATUS EQL CLIS_LOCNEG
327 0432 2 THEN
328 0433 3 BEGIN
329 0434 3 INCR I FROM 2 TO ACTUALCOUNT() DO
330 0435 4 BEGIN
331 0436 4 Q_ICURSOR[0,0,16,0] = 0;
332 0437 4 Q_ICURSOR[2,0,16,0] = ACTUALPARAMETER(.I);
333 0438 4 Q_ICURSOR[4,0,32,0] = 0;
334 0439 4 Q_ICURSOR[8,0,32,0] = 0;
335 0440 4 Q_ICURSOR = .Q_ICURSOR + 12;
336 0441 3 END;
337 0442 2 END;
338 0443 1 END;
```

0000 00000

.ENTRY PARSE\_IF\_FALSE, Save nothing

: 0393



00000000G	00	04	AC	DD	00002	PUSHL	Q, DESC	:	0430
00000000G	8F		01	FB	00005	CALLS	#T, CLISPRESNT	:	
			50	D1	0000C	CMPL	STATUS, #CLIS_NEGATED	:	0431
00000000G	8F		09	13	00013	BEQL	1\$	:	
			50	D1	00015	CMPL	STATUS, #CLIS_LOCNEG	:	
			14	12	0001C	BNEQ	4\$	:	
	51		6C	9A	0001E	MOVZBL	(AP), R1	:	0434
	50		01	D0	00021	MOVL	#1, I	:	
			08	11	00024	BRB	3\$	:	
			8B	B4	00026	CLRW	(Q_ICURSOR)+	:	0436
	8B		6C40	F7	00028	CVTLW	(AP)[I], (Q_ICURSOR)+	:	0437
			8B	7C	0002C	CLRQ	(Q_ICURSOR)+	:	0438
F4	50		51	F3	0002E	AOBLEQ	R1, I, 2\$	:	0434
			04	00	00032	RET		:	0443

; Routine Size: 51 bytes, Routine Base: CODE + 0052



```
340 0444 1 GLOBAL ROUTINE PARSE_IF_TRUE_FALSE(PARSE_PARAMETERS_): PARSE_LINKAGE=
341 0445 1
342 0446 1 !++
343 0447 1
344 0448 1 FUNCTIONAL DESCRIPTION:
345 0449 1 This routine parses a Boolean qualifier, making an entry in the job
346 0450 1 controller parameter list according to the true or false status.
347 0451 1
348 0452 1 INPUT PARAMETERS:
349 0453 1 Standard parser parameters.
350 0454 1
351 0455 1 IMPLICIT INPUTS:
352 0456 1 NONE
353 0457 1
354 0458 1 OUTPUT PARAMETERS:
355 0459 1 NONE
356 0460 1
357 0461 1 IMPLICIT OUTPUTS:
358 0462 1 NONE
359 0463 1
360 0464 1 ROUTINE VALUE:
361 0465 1 NONE
362 0466 1
363 0467 1 SIDE EFFECTS:
364 0468 1 NONE
365 0469 1
366 0470 1 !--
367 0471 1
368 0472 2 BEGIN
369 0473 2 PARSE_EXTERNAL_REGISTERS; ! Declare external registers
370 0474 2 LOCAL
371 0475 2 STATUS;
372 0476 2
373 0477 2
374 0478 2 STATUS = CLIS$PRESENT(.Q_DESC);
375 0479 2 IF .STATUS
376 0480 2 THEN
377 0481 2 BEGIN
378 0482 2 Q_ICURSOR[0,0,16,0] = 0;
379 0483 2 Q_ICURSOR[2,0,16,0] = .Q_P1;
380 0484 2 Q_ICURSOR[4,0,32,0] = 0;
381 0485 2 Q_ICURSOR[8,0,32,0] = 0;
382 0486 2 Q_ICURSOR = .Q_ICURSOR + 12;
383 0487 2 END
384 0488 2 ELSE IF .STATUS EQL CLIS$_NEGATED OR .STATUS EQL CLIS$_LOCNEG
385 0489 2 THEN
386 0490 2 BEGIN
387 0491 2 Q_ICURSOR[0,0,16,0] = 0;
388 0492 2 Q_ICURSOR[2,0,16,0] = .Q_P2;
389 0493 2 Q_ICURSOR[4,0,32,0] = 0;
390 0494 2 Q_ICURSOR[8,0,32,0] = 0;
391 0495 2 Q_ICURSOR = .Q_ICURSOR + 12;
392 0496 2 END;
393 0497 1 END;
```



			0000	00000		.ENTRY	PARSE_IF_TRUE_FALSE, Save nothing	:	0444
		04	AC	DD 00002		PUSHL	Q_DESC	:	0478
00000000G	00		01	FB 00005		CALLS	#T, CLISPRESNT	:	
	09		50	E9 0000C		BLBC	STATUS, 1\$	:	0479
			6B	B4 0000F		CLRW	(Q_ICURSOR)	:	0482
02	AB	08	AC	B0 00011		MOVW	Q_P1, 2(Q_ICURSOR)	:	0483
			19	11 00016		BRB	3\$	:	0484
00000000G	8F		50	D1 00018	1\$:	CMPL	STATUS, #CLIS_NEGATED	:	0488
			09	13 0001F		BEQL	2\$	:	
00000000G	8F		50	D1 00021		CMPL	STATUS, #CLIS_LOCNEG	:	
			0D	12 00028		BNEQ	4\$	:	
			6B	B4 0002A	2\$:	CLRW	(Q_ICURSOR)	:	0491
02	AB	0C	AC	B0 0002C		MOVW	Q_P2, 2(Q_ICURSOR)	:	0492
		04	AB	7C 00031	3\$:	CLRW	4(Q_ICURSOR)	:	0493
	5B		0C	C0 00034		ADDL2	#12, Q_ICURSOR	:	0495
			04	00037	4\$:	RET		:	0497

; Routine Size: 56 bytes, Routine Base: CODE + 0085



```
395 0498 1 GLOBAL ROUTINE PARSE_LOCAL_TRUE_FALSE(PARSE_PARAMETERS_): PARSE_LINKAGE=
396 0499 1
397 0500 1 ++
398 0501 1
399 0502 1 FUNCTIONAL DESCRIPTION:
400 0503 1 This routine parses a Boolean qualifier, making an entry in the job
401 0504 1 controller parameter list according to the locally present or negated
402 0505 1 status.
403 0506 1
404 0507 1 INPUT PARAMETERS:
405 0508 1 Standard parser parameters.
406 0509 1
407 0510 1 IMPLICIT INPUTS:
408 0511 1 NONE
409 0512 1
410 0513 1 OUTPUT PARAMETERS:
411 0514 1 NONE
412 0515 1
413 0516 1 IMPLICIT OUTPUTS:
414 0517 1 NONE
415 0518 1
416 0519 1 ROUTINE VALUE:
417 0520 1 NONE
418 0521 1
419 0522 1 SIDE EFFECTS:
420 0523 1 NONE
421 0524 1
422 0525 1 --
423 0526 1
424 0527 2 BEGIN
425 0528 2 PARSE_EXTERNAL_REGISTERS; ! Declare external registers
426 0529 2 LOCAL
427 0530 2 STATUS;
428 0531 2
429 0532 2
430 0533 2 STATUS = CLISPRESNT(.Q DESC);
431 0534 2 IF .STATUS EQL CLIS_LOCPRES
432 0535 2 THEN
433 0536 2 BEGIN
434 0537 2 Q_ICURSOR[0,0,16,0] = 0;
435 0538 2 Q_ICURSOR[2,0,16,0] = .Q_P1;
436 0539 2 Q_ICURSOR[4,0,32,0] = 0;
437 0540 2 Q_ICURSOR[8,0,32,0] = 0;
438 0541 2 Q_ICURSOR = .Q_ICURSOR + 12;
439 0542 2 END
440 0543 2 ELSE IF .STATUS EQL CLIS_LOCNEG
441 0544 2 THEN
442 0545 2 BEGIN
443 0546 2 Q_ICURSOR[0,0,16,0] = 0;
444 0547 2 Q_ICURSOR[2,0,16,0] = .Q_P2;
445 0548 2 Q_ICURSOR[4,0,32,0] = 0;
446 0549 2 Q_ICURSOR[8,0,32,0] = 0;
447 0550 2 Q_ICURSOR = .Q_ICURSOR + 12;
448 0551 2 END;
449 0552 1 END;
```



			0000 00000	.ENTRY	PARSE_LOCAL_TRUE_FALSE, Save nothing	: 0498
		04	AC DD 00002	PUSHL	Q_DESC	: 0533
00000000G	00		01 FB 00005	CALLS	#T, CLISPRES	:
00000000G	8F		50 D1 0000C	CMPL	STATUS, #CLIS_LOCPRES	: 0534
			09 12 00013	BNEQ	1\$	:
			6B B4 00015	CLRW	(Q_ICURSOR)	: 0537
02	AB	08	AC B0 00017	MOVW	Q_P1, 2(Q_ICURSOR)	: 0538
			10 11 0001C	BRB	2\$	: 0539
00000000G	8F		50 D1 0001E 1\$:	CMPL	STATUS, #CLIS_LOCNEG	: 0543
			0D 12 00025	BNEQ	3\$	:
			6B B4 00027	CLRW	(Q_ICURSOR)	: 0546
02	AB	0C	AC B0 00029	MOVW	Q_P2, 2(Q_ICURSOR)	: 0547
		04	AB 7C 0002E 2\$:	CLRQ	4(Q_ICURSOR)	: 0548
	5B		0C C0 00031	ADDL2	#12, Q_ICURSOR	: 0550
			04 00034 3\$:	RET		: 0552

; Routine Size: 53 bytes, Routine Base: CODE + 00BD



```
451 0553 1 GLOBAL ROUTINE PARSE_AFTER(PARSE_PARAMETERS_): PARSE_LINKAGE=
452 0554 1
453 0555 1 !++
454 0556 1
455 0557 1 FUNCTIONAL DESCRIPTION:
456 0558 1 This routine parses the /AFTER qualifier, making an entry in the job
457 0559 1 controller parameter list.
458 0560 1
459 0561 1 INPUT PARAMETERS:
460 0562 1 Standard parser parameters.
461 0563 1
462 0564 1 IMPLICIT INPUTS:
463 0565 1 NONE
464 0566 1
465 0567 1 OUTPUT PARAMETERS:
466 0568 1 NONE
467 0569 1
468 0570 1 IMPLICIT OUTPUTS:
469 0571 1 NONE
470 0572 1
471 0573 1 ROUTINE VALUE:
472 0574 1 NONE
473 0575 1
474 0576 1 SIDE EFFECTS:
475 0577 1 NONE
476 0578 1
477 0579 1 !--
478 0580 1
479 0581 2 BEGIN
480 0582 2 PARSE_EXTERNAL_REGISTERS; ! Declare external registers
481 0583 2 LOCAL
482 0584 2 STATUS;
483 0585 2
484 0586 2
485 0587 2 STATUS = CLIS$PRESENT(.Q_DESC);
486 0588 2 IF .STATUS
487 0589 2 THEN
488 0590 3 BEGIN
489 0591 3 CLIS$GET VALUE(.Q_DESC, .Q_VALUE_DESC);
490 0592 3 IF NOT [IB$CVT_TIME(.Q_VALUE_DESC, .Q_DCUSOR)]
491 0593 3 THEN
492 0594 3 SIGNAL_STOP(.Q_MESSAGE, 2, .Q_VALUE_DESC, .Q_DESC);
493 0595 3
494 0596 3
495 0597 3 Q_ICUSOR[0,0,16,0] = 8;
496 0598 3 Q_ICUSOR[2,0,16,0] = SJC$ AFTER_TIME;
497 0599 3 Q_ICUSOR[4,0,32,0] = .Q_DCUSOR;
498 0600 3 Q_ICUSOR[8,0,32,0] = 0;
499 0601 3 Q_ICUSOR = .Q_ICUSOR + 12;
500 0602 3 Q_DCUSOR = .Q_DCUSOR + 8;
501 0603 3 END;
502 0604 2 IF .STATUS EQL CLIS$_NEGATED OR .STATUS EQL CLIS$_LOCNEG
503 0605 2 THEN
504 0606 3 BEGIN
505 0607 3 Q_ICUSOR[0,0,16,0] = 0;
506 0608 3 Q_ICUSOR[2,0,16,0] = SJC$_NO_AFTER_TIME;
507 0609 3 Q_ICUSOR[4,0,32,0] = 0;
```



```
: 508
: 509
: 510
: 511

0610 3      Q_ICursor[8,0,32,0] = 0;
0611 3      Q_ICursor = .Q_ICursor + 12;
0612 2      END;
0613 1 END;
```

			0004 00000	.ENTRY	PARSE AFTER, Save R2		0553
		04	AC DD 00002	PUSHL	Q_DEST		0587
00000000G	00		01 FB 00005	CALLS	#T, CLISPRESENT		
	52		50 D0 0000C	MOVL	R0, STATUS		
	36		52 E9 00C0F	BLBC	STATUS, 2\$		0588
		04	58 DD 00012	PUSHL	Q_VALUE_DESC		0591
			AC DD 00014	PUSHL	Q_DESC		
00000000G	00		02 FB 00017	CALLS	#2, CLISGET_VALUE		
		0500	8F BB 0001E	PUSHR	#^M<R8,R10>		0592
00000000G	00		02 FB 00022	CALLS	#2, LIB\$CVT_TIME		
	10		50 E8 00029	BLBS	R0, 1\$		
		04	AC DD 0002C	PUSHL	Q_DESC		0594
			58 DD 0002F	PUSHL	Q_VALUE_DESC		
			02 DD 00031	PUSHL	#2		
			59 DD 00033	PUSHL	Q_MESSAGE		
00000000G	00		04 FB 00035	CALLS	#Z, LIB\$STOP		
	8B 00030008		8F D0 0003C 1\$:	MOVL	#196616, (Q_ICursor)+		0597
	8B		8A 7E 00043	MOVAQ	(Q_ICursor)+, (Q_ICursor)+		0599
			8B D4 00046	CLRL	(Q_ICursor)+		0600
00000000G	8F		52 D1 00048 2\$:	CMPL	STATUS, #CLIS_NEGATED		0604
			09 13 0004F	BEQL	3\$		
00000000G	8F		52 D1 00051	CMPL	STATUS, #CLIS_LOCNEG		
			09 12 00058	BNEQ	4\$		
	8B 00040000		8F D0 0005A 3\$:	MOVL	#262144, (Q_ICursor)+		0607
			8B 7C 00061	CLRQ	(Q_ICursor)+		0609
			04 00063 4\$:	RET			0613

; Routine Size: 100 bytes, Routine Base: CODE + 00F2



```
513 0614 1 GLOBAL ROUTINE PARSE_ALIGN(PARSE_PARAMETERS_): PARSE_LINKAGE=
514 0615 1
515 0616 1 ++
516 0617 1
517 0618 1 FUNCTIONAL DESCRIPTION:
518 0619 1 This routine parses the /ALIGN qualifier, making an entry in the job
519 0620 1 controller parameter list.
520 0621 1
521 0622 1 INPUT PARAMETERS:
522 0623 1 Standard parser parameters.
523 0624 1
524 0625 1 IMPLICIT INPUTS:
525 0626 1 NONE
526 0627 1
527 0628 1 OUTPUT PARAMETERS:
528 0629 1 NONE
529 0630 1
530 0631 1 IMPLICIT OUTPUTS:
531 0632 1 NONE
532 0633 1
533 0634 1 ROUTINE VALUE:
534 0635 1 NONE
535 0636 1
536 0637 1 SIDE EFFECTS:
537 0638 1 NONE
538 0639 1
539 0640 1 --
540 0641 1
541 0642 2 BEGIN
542 0643 2 PARSE_EXTERNAL_REGISTERS; ! Declare external registers
543 0644 2 LOCAL
544 0645 2 EXPLICIT_PAGES;
545 0646 2
546 0647 2
547 0648 2 IF CLISPRESNT(.Q_DESC)
548 0649 2 THEN
549 0650 3 BEGIN
550 0651 3 EXPLICIT_PAGES = FALSE;
551 0652 3
552 0653 3
553 0654 3 WHILE CLISGET_VALUE(.Q_DESC, .Q_VALUE_DESC) DO
554 0655 4 BEGIN
555 0656 4 IF LIB$CVT DTB(
556 0657 4 .Q_VALUE_DESC[DSC$W_LENGTH], .Q_VALUE_DESC[DSC$A_POINTER],
557 0658 4 .Q_DCUSOR)
558 0659 4 THEN
559 0660 5 BEGIN
560 0661 5 IF .Q_DCUSOR[0,0,32,0] - 1 GTRU 20 - 1 ! 1 <= N <= 20
561 0662 5 THEN
562 0663 5 SIGNAL_STOP(.Q_MESSAGE, 2, .Q_VALUE_DESC, .Q_DESC);
563 0664 5
564 0665 5
565 0666 5 EXPLICIT_PAGES = TRUE;
566 0667 5 Q_ICUSOR[0,0,16,0] = 4;
567 0668 5 Q_ICUSOR[2,0,16,0] = SJC$ ALIGNMENT_PAGES;
568 0669 5 Q_ICUSOR[4,0,32,0] = .Q_DCUSOR;
569 0670 5 Q_ICUSOR[8,0,32,0] = 0;
```



```
570      Q_ICURSOR = .Q_ICURSOR + 12;
571      Q_DCURSOR = .Q_DCURSOR + 4;
572      END
573
574      ELSE IF CALL_TPARE(.Q_VALUE_DESC, MASK_STATES, MASK_KEYS)
575      THEN
576      BEGIN
577      Q_ICURSOR[0,0,16,0] = 0;
578      Q_ICURSOR[2,0,16,0] = SJCS_ALIGNMENT_MASK;
579      Q_ICURSOR[4,0,32,0] = 0;
580      Q_ICURSOR[8,0,32,0] = 0;
581      Q_ICURSOR = .Q_ICURSOR + 12;
582      END
583
584      ELSE
585      SIGNAL_STOP(.Q_MESSAGE, 2, .Q_VALUE_DESC, .Q_DESC);
586      END;
587
588      IF NOT .EXPLICIT_PAGES
589      THEN
590      BEGIN
591      Q_ICURSOR[0,0,16,0] = 4;
592      Q_ICURSOR[2,0,16,0] = SJCS_ALIGNMENT_PAGES;
593      Q_ICURSOR[4,0,32,0] = LITERAL_ONE;
594      Q_ICURSOR[8,0,32,0] = 0;
595      Q_ICURSOR = .Q_ICURSOR + 12;
596      END;
597      END;
598      END;
599      END;
600      END;
601      END;
```

			000C 00000	.ENTRY	PARSE ALIGN, Save R2,R3	0614
	53	00000000G	00 9E 00002	MOVAB	LIB\$STOP, R3	0648
		04	AC DD 00009	PUSHL	Q_DESC	
00000000G	00		01 FB 0000C	CALLS	#T, CL\$PRESENT	
	01		50 E8 00013	BLBS	R0, 1\$	
			04 00016	RET		
			52 D4 00017 1\$:	CLRL	EXPLICIT_PAGES	0651
			58 DD 00019 2\$:	PUSHL	Q_VALUE_DESC	0654
		04	AC DD 0001B	PUSHL	Q_DESC	
00000000G	00		02 FB 0001E	CALLS	#2, CL\$GET_VALUE	
	63		50 E9 00025	BLBC	R0, 6\$	
			5A DD 00028	PUSHL	Q_DCURSOR	0658
		04	A8 DD 0002A	PUSHL	4Q_VALUE_DESC	0657
	7E		68 3C 0002D	MOVZWL	(Q_VALUE_DESC), -(SP)	
00000000G	00		03 FB 00030	CALLS	#3, LIB\$CVT_DTB	
	26		50 E9 00037	BLBC	R0, 4\$	
50	6A		01 C3 0003A	SUBL3	#1, (Q_DCURSOR), R0	0661
	13		50 D1 0003E	CMPL	R0, #19	
			0C 1B 00041	BLEQU	3\$	
		04	AC DD 00043	PUSHL	Q_DESC	0663



		58	DD	00046	PUSHL	Q_VALUE_DESC	:
		02	DD	00048	PUSHL	#2	:
		59	DD	0004A	PUSHL	Q_MESSAGE	:
63		04	FB	0004C	CALLS	#7, LIB\$STOP	:
52		01	D0	0004F	3\$:	MOVL	#1, EXPLICIT_PAGES
8B	00060004	8F	D0	00052	MOVL	#393220, (Q_ICURSOR)+	0666
8B		8A	DE	00059	MOVAL	(Q_DCURSOR)+, (Q_ICURSOR)+	0667
		8B	D4	0005C	CLRL	(Q_ICURSOR)+	0669
		B9	11	0005E	BRB	2\$	0670
	0000V	CF	9F	00060	4\$:	PUSHAB	MASK_KEYS
	0000V	CF	9F	00064	PUSHAB	MASK_STATES	0676
		58	DD	00068	PUSHL	Q_VALUE_DESC	:
FE47	CF	03	FB	0006A	CALLS	#3, CALL_TPARSE	:
	0B	50	E9	0006F	BLBC	R0, 5\$	:
	8B	8F	D0	00072	MOVL	#327680, (Q_ICURSOR)+	0679
		8B	7C	00079	CLRQ	(Q_ICURSOR)+	0681
		9C	11	0007B	BRB	2\$	0676
		AC	DD	0007D	5\$:	PUSHL	Q_DESC
	04	58	DD	00080	PUSHL	Q_VALUE_DESC	0688
		02	DD	00082	PUSHL	#2	:
		59	DD	00084	PUSHL	Q_MESSAGE	:
63		04	FB	00086	CALLS	#7, LIB\$STOP	:
		8E	11	00089	BRB	2\$	0654
		52	E8	0008B	6\$:	BLBS	EXPLICIT_PAGES, 7\$
	0E	8F	D0	0008E	MOVL	#393220, (Q_ICURSOR)+	0692
	8B	CF	9E	00095	MOVAB	LITERAL ONE, (Q_ICURSOR)+	0695
	8B	D4	0009A	CLRL	(Q_ICURSOR)+	0697	0698
		04	0009C	7\$:	RET		0702

; Routine Size: 157 bytes, Routine Base: CODE + 0156



```
603 0703 1 GLOBAL ROUTINE PARSE_BACKWARD(PARSE_PARAMETERS_): PARSE_LINKAGE=
604 0704 1
605 0705 1 ++
606 0706 1
607 0707 1 FUNCTIONAL DESCRIPTION:
608 0708 1 This routine parses the /BACKWARD qualifier, making an entry in the
609 0709 1 job controller parameter list.
610 0710 1
611 0711 1 INPUT PARAMETERS:
612 0712 1 Standard parser parameters.
613 0713 1
614 0714 1 IMPLICIT INPUTS:
615 0715 1 NONE
616 0716 1
617 0717 1 OUTPUT PARAMETERS:
618 0718 1 NONE
619 0719 1
620 0720 1 IMPLICIT OUTPUTS:
621 0721 1 NONE
622 0722 1
623 0723 1 ROUTINE VALUE:
624 0724 1 NONE
625 0725 1
626 0726 1 SIDE EFFECTS:
627 0727 1 NONE
628 0728 1
629 0729 1 --
630 0730 1
631 0731 2 BEGIN
632 0732 2 PARSE_EXTERNAL_REGISTERS; ! Declare external registers
633 0733 2
634 0734 2
635 0735 2 IF CLISPRESNT(.Q_DESC)
636 0736 2 THEN
637 0737 3 BEGIN
638 0738 3 IF CLISGET_VALUE(.Q_DESC, .Q_VALUE_DESC)
639 0739 3 THEN
640 0740 4 BEGIN
641 0741 4 IF
642 0742 5 BEGIN
643 0743 5 IF NOT LIB$CVT DTB(
644 0744 5 .Q_VALUE_DESC[DSC$W_LENGTH], .Q_VALUE_DESC[DSC$A_POINTER],
645 0745 5 .Q_DCUSOR)
646 0746 5 THEN
647 0747 5 TRUE
648 0748 5 ELSE
649 0749 5 .Q_DCUSOR[0,0,32,0] EQL 0
650 0750 5 END
651 0751 4 THEN
652 0752 4 SIGNAL_STOP(.Q_MESSAGE, 2, .Q_VALUE_DESC, .Q_DESC);
653 0753 4
654 0754 4
655 0755 4 Q_DCUSOR[0,0,32,0] = - .Q_DCUSOR[0,0,32,0];
656 0756 4 Q_ICUSOR[0,0,16,0] = 4;
657 0757 4 Q_ICUSOR[2,0,16,0] = SJCS_RELATIVE_PAGE;
658 0758 4 Q_ICUSOR[4,0,32,0] = .Q_DCUSOR;
659 0759 4 Q_ICUSOR[8,0,32,0] = 0;
```



```
: 660      0760 4      Q_ICursor = .Q_ICursor + 12;
: 661      0761 4      Q_DCursor = .Q_DCursor + 4;
: 662      0762 4      END
: 663      0763 3      ELSE
: 664      0764 4      BEGIN
: 665      0765 4      Q_ICursor[0,0,16,0] = 4;
: 666      0766 4      Q_ICursor[2,0,16,0] = SJCS_RELATIVE_PAGE;
: 667      0767 4      Q_ICursor[4,0,32,0] = LITERAL_MINUS_ONE;
: 668      0768 4      Q_ICursor[8,0,32,0] = 0;
: 669      0769 4      Q_ICursor = .Q_ICursor + 12;
: 670      0770 3      END;
: 671      0771 2      END;
: 672      0772 1      END;
```

			0000 00000	.ENTRY	PARSE BACKWARD, Save nothing	: 0703
		04	AC DD 00002	PUSHL	Q_DESC	: 0735
00000000G	00		01 FB 00005	CALLS	#1, CLISPRESNT	
	58		50 E9 0000C	BLBC	R0, 5\$	
		04	58 DD 0000F	PUSHL	Q_VALUE_DESC	: 0738
00000000G	00		AC DD 00011	PUSHL	Q_DESC	
	36		02 FB 00014	CALLS	#2, CLISGET_VALUE	
			50 E9 0001B	BLBC	R0, 3\$	
		04	5A DD 0001E	PUSHL	Q_DCursor	: 0745
			A8 DD 00020	PUSHL	4(Q_VALUE_DESC)	: 0744
00000000G	7E		68 3C 00023	MOVZWL	(Q_VALUE_DESC), -(SP)	
	00		03 FB 00026	CALLS	#3, LIB\$CVT_DTB	
	04		50 E9 0002D	BLBC	R0, 1\$	
			6A D5 00030	TSTL	(Q_DCursor)	: 0749
		04	10 12 00032	BNEQ	2\$	
			AC DD 00034 1\$:	PUSHL	Q_DESC	: 0752
			58 DD 00037	PUSHL	Q_VALUE_DESC	
			02 DD 00039	PUSHL	#2	
00000000G	00		59 DD 0003B	PUSHL	Q_MESSAGE	
	6A		04 FB 0003D	CALLS	#4, LIB\$STOP	
	6B 00880004		6A CE 00044 2\$:	MNEGL	(Q_DCursor), (Q_DCursor)	: 0755
	04 AB		8F D0 00047	MOVL	#8912900, (Q_ICursor)	: 0756
			8A DE 0004E	MOVAL	(Q_DCursor)+, 4(Q_ICursor)	: 0758
			0D 11 00052	BRB	4\$	: 0759
	6B 00880004		8F D0 00054 3\$:	MOVL	#8912900, (Q_ICursor)	: 0765
	04 AB FDAE		CF 9E 0005B	MOVAB	LITERAL_MINUS_ONE, 4(Q_ICursor)	: 0767
			AB D4 00061 4\$:	CLRL	8(Q_ICursor)	: 0768
	5B		0C C0 00064	ADDL2	#12, Q_ICursor	: 0760
			04 00067 5\$:	RET		: 0772

; Routine Size: 104 bytes, Routine Base: CODE + 01F3



```

674 0773 1 GLOBAL ROUTINE PARSE_BASE_PRIORITY(PARSE_PARAMETERS_): PARSE_LINKAGE=
675 0774 1
676 0775 1 !++
677 0776 1
678 0777 1 FUNCTIONAL DESCRIPTION:
679 0778 1 This routine parses the /BASE_PRIORITY qualifier, making an entry in the
680 0779 1 job controller parameter list.
681 0780 1
682 0781 1 INPUT PARAMETERS:
683 0782 1 Standard parser parameters.
684 0783 1
685 0784 1 IMPLICIT INPUTS:
686 0785 1 NONE
687 0786 1
688 0787 1 OUTPUT PARAMETERS:
689 0788 1 NONE
690 0789 1
691 0790 1 IMPLICIT OUTPUTS:
692 0791 1 NONE
693 0792 1
694 0793 1 ROUTINE VALUE:
695 0794 1 NONE
696 0795 1
697 0796 1 SIDE EFFECTS:
698 0797 1 NONE
699 0798 1
700 0799 1 !--
701 0800 1
702 0801 2 BEGIN
703 0802 2 PARSE_EXTERNAL_REGISTERS; ! Declare external registers
704 0803 2
705 0804 2
706 0805 2 IF CLISGET_VALUE(.Q_DESC, .Q_VALUE_DESC)
707 0806 2 THEN
708 0807 3 BEGIN
709 0808 3 IF
710 0809 4 BEGIN
711 0810 4 IF NOT LIB$CVT DTB(
712 0811 4 .Q_VALUE_DESC[DSC$W_LENGTH], .Q_VALUE_DESC[DSC$A_POINTER],
713 0812 4 .Q_DCUSOR)
714 0813 4 THEN
715 0814 4 TRUE
716 0815 4 ELSE
717 0816 4 .Q_DCUSOR[0,0,32,0] GTRU 15 ! 0 <= N <= 15
718 0817 4 END
719 0818 3 THEN
720 0819 3 SIGNAL_STOP(.Q_MESSAGE, 2, .Q_VALUE_DESC, .Q_DESC);
721 0820 3
722 0821 3
723 0822 3 Q_ICUSOR[0,0,16,0] = 4;
724 0823 3 Q_ICUSOR[2,0,16,0] = SJC$ BASE_PRIORITY;
725 0824 3 Q_ICUSOR[4,0,32,0] = .Q_DCUSOR;
726 0825 3 Q_ICUSOR[8,0,32,0] = 0;
727 0826 3 Q_ICUSOR = .Q_ICUSOR + 12;
728 0827 3 Q_DCUSOR = .Q_DCUSOR + 4;
729 0828 2 END;
730 0829 1 END;
```



			0000 00000	.ENTRY	PARSE BASE PRIORITY, Save nothing	: 0773
			58 DD 00002	PUSHL	Q_VALUE_DESC	: 0805
		04	AC DD 00004	PUSHL	Q_DESC	:
00000000G	00		02 FB 00007	CALLS	#2, CLISGET_VALUE	:
	33		50 E9 0000E	BLBC	R0, 3\$	:
			5A DD 00011	PUSHL	Q_DCURSOR	: 0812
		04	A8 DD 00013	PUSHL	4(Q_VALUE_DESC)	: 0811
	7E		68 3C 00016	MOVZWL	(Q_VALUE_DESC), -(SP)	:
00000000G	00		03 FB 00019	CALLS	#3, LIB\$CVT_DTB	:
	05		50 E9 00020	BLBC	R0, 1\$	:
	0F		6A D1 00023	CMPL	(Q_DCURSOR), #15	: 0816
			10 1B 00026	BLEQU	2\$	:
		04	AC DD 00028 1\$:	PUSHL	Q_DESC	: 0819
			58 DD 0002B	PUSHL	Q_VALUE_DESC	:
			02 DD 0002D	PUSHL	#2	:
			59 DD 0002F	PUSHL	Q_MESSAGE	:
00000000G	00		04 FB 00031	CALLS	#4, LIB\$STOP	:
	8B 00070004		8F D0 00038 2\$:	MOVL	#458756, (Q_ICURSOR)+	: 0822
	8B		8A DE 0003F	MOVAL	(Q_DCURSOR)+, (Q_ICURSOR)+	: 0824
			8B D4 00042	CLRL	(Q_ICURSOR)+	: 0825
			04 00044 3\$:	RET		: 0829

; Routine Size: 69 bytes, Routine Base: CODE + 025B



```
732 0830 1 GLOBAL ROUTINE PARSE_BUFFER_COUNT(PARSE_PARAMETERS_): PARSE_LINKAGE=
733 0831 1
734 0832 1 !++
735 0833 1
736 0834 1 FUNCTIONAL DESCRIPTION:
737 0835 1 This routine parses the /BUFFER_COUNT qualifier, making an entry in the
738 0836 1 job controller parameter list.
739 0837 1
740 0838 1 INPUT PARAMETERS:
741 0839 1 Standard parser parameters.
742 0840 1
743 0841 1 IMPLICIT INPUTS:
744 0842 1 NONE
745 0843 1
746 0844 1 OUTPUT PARAMETERS:
747 0845 1 NONE
748 0846 1
749 0847 1 IMPLICIT OUTPUTS:
750 0848 1 NONE
751 0849 1
752 0850 1 ROUTINE VALUE:
753 0851 1 NONE
754 0852 1
755 0853 1 SIDE EFFECTS:
756 0854 1 NONE
757 0855 1
758 0856 1 !--
759 0857 1
760 0858 2 BEGIN
761 0859 2 PARSE_EXTERNAL_REGISTERS; ! Declare external registers
762 0860 2
763 0861 2
764 0862 2 IF CLISGET_VALUE(.Q_DESC, .Q_VALUE_DESC)
765 0863 2 THEN
766 0864 3 BEGIN
767 0865 3 IF
768 0866 4 BEGIN
769 0867 4 IF NOT LIB$CVT_DTB(
770 0868 4 .Q_VALUE_DESC[DSC$W_LENGTH], .Q_VALUE_DESC[DSC$A_POINTER],
771 0869 4 .Q_DCUSOR)
772 0870 4 THEN
773 0871 4 TRUE
774 0872 4 ELSE
775 0873 4 .Q_DCUSOR[0,0,32,0] GTRU 127 ! 0 <= N <= 127
776 0874 4 END
777 0875 3 THEN
778 0876 3 SIGNAL_STOP(.Q_MESSAGE, 2, .Q_VALUE_DESC, .Q_DESC);
779 0877 3
780 0878 3
781 0879 3 Q_ICUSOR[0,0,16,0] = 4;
782 0880 3 Q_ICUSOR[2,0,16,0] = SJCS_BUFFER_COUNT;
783 0881 3 Q_ICUSOR[4,0,32,0] = .Q_DCUSOR;
784 0882 3 Q_ICUSOR[8,0,32,0] = 0;
785 0883 3 Q_ICUSOR = .Q_ICUSOR + 12;
786 0884 3 Q_DCUSOR = .Q_DCUSOR + 4;
787 0885 2 END;
788 0886 1 END;
```



			0000 00000	.ENTRY	PARSE BUFFER_COUNT, Save nothing	: 0830
			58 DD 00002	PUSHL	Q_VALUE_DESC	: 0862
		04	AC DD 00004	PUSHL	Q_DESC	:
00000000G	00		02 FB 00007	CALLS	#2, CLISGET_VALUE	:
	37		50 E9 0000E	BLBC	R0, 3\$	:
			5A DD 00011	PUSHL	Q_DCURSOR	: 0869
		04	A8 DD 00013	PUSHL	4(Q_VALUE_DESC)	: 0868
	7E		68 3C 00016	MOVZWL	(Q_VALUE_DESC), -(SP)	:
00000000G	00		03 FB 00019	CALLS	#3, LIB\$CVT_DTB	:
	09		50 E9 00020	BLBC	R0, 1\$	:
0000007F	8F		6A D1 00023	CMPL	(Q_DCURSOR), #127	: 0873
			10 1B 0002A	BLEQU	2\$	:
		04	AC DD 0002C 1\$:	PUSHL	Q_DESC	: 0876
			58 DD 0002F	PUSHL	Q_VALUE_DESC	:
			02 DD 00031	PUSHL	#2	:
			59 DD 00033	PUSHL	Q_MESSAGE	:
00000000G	00		04 FB 00035	CALLS	#4, LIB\$STOP	:
	8B 00A00004		8F D0 0003C 2\$:	MOVL	#10485764, (Q_ICURSOR)+	: 0879
	8B		8A DE 00043	MOVAL	(Q_DCURSOR)+, (Q_ICURSOR)+	: 0881
			8B D4 00046	CLRL	(Q_ICURSOR)+	: 0882
			04 00048 3\$:	RET		: 0886

; Routine Size: 73 bytes, Routine Base: CODE + 02A0



```
790 0887 1 GLOBAL ROUTINE PARSE_CHARACTERISTICS(PARSE_PARAMETERS_): PARSE_LINKAGE=
791 0888 1
792 0889 1 |++
793 0890 1 |
794 0891 1 | FUNCTIONAL DESCRIPTION:
795 0892 1 |     This routine parses the /CHARACTERISTICS qualifier, making an entry in
796 0893 1 |     the job controller parameter list.
797 0894 1 |
798 0895 1 | INPUT PARAMETERS:
799 0896 1 |     Standard parser parameters.
800 0897 1 |
801 0898 1 | IMPLICIT INPUTS:
802 0899 1 |     NONE
803 0900 1 |
804 0901 1 | OUTPUT PARAMETERS:
805 0902 1 |     NONE
806 0903 1 |
807 0904 1 | IMPLICIT OUTPUTS:
808 0905 1 |     NONE
809 0906 1 |
810 0907 1 | ROUTINE VALUE:
811 0908 1 |     NONE
812 0909 1 |
813 0910 1 | SIDE EFFECTS:
814 0911 1 |     NONE
815 0912 1 |
816 0913 1 | --
817 0914 1 |
818 0915 2 BEGIN
819 0916 2 PARSE_EXTERNAL_REGISTERS;      ! Declare external registers
820 0917 2 LOCAL
821 0918 2     STATUS;
822 0919 2
823 0920 2
824 0921 2 STATUS = CLISPRESENT(.Q_DESC);
825 0922 2 IF .STATUS
826 0923 2 THEN
827 0924 3     BEGIN
828 0925 3         WHILE CLISGET_VALUE(.Q_DESC, .Q_VALUE_DESC) DO
829 0926 4             BEGIN
830 0927 4                 IF LIB$CVT_DTB(
831 0928 4                     .Q_VALUE_DESC[DSC$W_LENGTH], .Q_VALUE_DESC[DSC$A_POINTER],
832 0929 4                     .Q_DCUSOR)
833 0930 4                 THEN
834 0931 5                     BEGIN
835 0932 5                         IF .Q_DCUSOR[0,0,32,0] GTRU 127      ! 0 <= N <= 127
836 0933 5                         THEN
837 0934 5                             SIGNAL_STOP(.Q_MESSAGE, 2, .Q_VALUE_DESC, .Q_DESC);
838 0935 5
839 0936 5
840 0937 5         Q_ICUSOR[0,0,16,0] = 4;
841 0938 5         Q_ICUSOR[2,0,16,0] = SJC$ CHARACTERISTIC_NUMBER;
842 0939 5         Q_ICUSOR[4,0,32,0] = .Q_DCUSOR;
843 0940 5         Q_ICUSOR[8,0,32,0] = 0;
844 0941 5         Q_ICUSOR = .Q_ICUSOR + 12;
845 0942 5         Q_DCUSOR = .Q_DCUSOR + 4;
846 0943 5     END
```



```

END;
IF .STATUS EQL CLIS_NEGATED OR .STATUS EQL CLIS_LOCNeg
THEN
  BEGIN
    Q_ICursor[0,0,16,0] = 0;
    Q_ICursor[2,0,16,0] = SJCS_NO_CHARACTERISTICS;
    Q_ICursor[4,0,32,0] = 0;
    Q_ICursor[8,0,32,0] = 0;
    Q_ICursor = .Q_ICursor + 12;
  END;
END;

```

00FC 00000				.ENTRY	PARSE_CHARACTERISTICS, Save R2,R3,R4,R5,R6,-;	0887
	57	00000000G	00 9E 00002	MOVAB	R7	:
		04	AC DD 00009	PUSHL	LIB\$STOP, R7	:
00000000G	00		01 FB 0000C	CALLS	Q_DESC	0921
	56		50 D0 00013	CALLS	#1, CLIS\$PRESENT	:
	79		56 E9 00016	MOVL	R0, STATUS	:
			58 DD 00019	BLBC	STATUS, 5\$	0922
			1\$: 58 DD 00019	PUSHL	Q_VALUE_DESC	0925
		04	AC DD 0001B	PUSHL	Q_DESC	:
00000000G	00		02 FB 0001E	CALLS	#2, CLIS\$GET_VALUE	:
	6A		50 E9 00025	BLBC	R0, 5\$	:
			5A DD 00028	PUSHL	Q_DCURSOR	0929
		04	A8 DD 0002A	PUSHL	4(Q_VALUE_DESC)	0928
	7E		68 3C 0002D	MOVZWL	(Q_VALUE_DESC), -(SP)	:
00000000G	00		03 FB 00030	CALLS	#3, LIB\$CVT_DTB	:
	23		50 E9 00037	BLBC	R0, 3\$	:
0000007F	8F		6A D1 0003A	CML	(Q_DCURSOR), #127	0932
			0C 1B 00041	BLEQU	2\$	:
		04	AC DD 00043	PUSHL	Q_DESC	0934
			58 DD 00046	PUSHL	Q_VALUE_DESC	:
			02 DD 00048	PUSHL	#2	:



		59	DD	0004A	PUSHL	Q MESSAGE	:	
	67	04	FB	0004C	CALLS	#4, LIB\$STOP	:	
	8B	8F	D0	0004F	2\$:	MOVL	#851972, (Q_ICURSOR)+	0937
	8B	8A	DE	00056	MOVAL	(Q_DCURSOR)+, (Q_ICURSOR)+	0939	
		8B	D4	00059	CLRL	(Q_ICURSOR)+	0940	
		BC	11	0005B	BRB	1\$	0927	
		0000V	CF	9F	3\$:	PUSHAB	SYMB_KEYS	0946
		0000V	CF	9F	PUSHAB	SYMB_STATES	:	
			58	DD	PUSHL	Q VALUE_DESC	:	
FCB7	CF	03	FB	00067	CALLS	#3, CALC_TPARSE	:	
	15	50	E9	0006C	BLBC	R0, 4\$	:	
	8B	68	B0	0006F	MOVW	(Q_VALUE_DESC), (Q_ICURSOR)+	0949	
	8B	0C	B0	00072	MOVW	#12, (Q_ICURSOR)+	0950	
	8B	5A	D0	00075	MOVL	Q_DCURSOR, (Q_ICURSOR)+	0951	
6A	04	8B	8B	D4	CLRL	(Q_ICURSOR)+	0952	
		68	28	0007A	MOVC3	(Q_VALUE_DESC), @4(Q_VALUE_DESC), -	0957	
						(Q_DCURSOR)	:	
	5A	53	D0	0007F	MOVL	R3, Q_DCURSOR	:	
		95	11	00082	BRB	1\$	0946	
		04	AC	DD	4\$:	PUSHL	Q_DESC	0962
			58	DD	PUSHL	Q_VALUE_DESC	:	
			02	DD	PUSHL	#2	:	
			59	DD	PUSHL	Q MESSAGE	:	
	67	04	FB	0008D	CALLS	#4, LIB\$STOP	:	
		87	11	00090	BRB	1\$	0925	
00000000G	8F	56	D1	00092	5\$:	CMPL	STATUS, #CLIS_NEGATED	0965
		09	13	00099	BEQL	6\$	:	
00000000G	8F	56	D1	0009B	CMPL	STATUS, #CLIS_LOCNEG	:	
		09	12	000A2	BNEQ	7\$	:	
	8B	8F	D0	000A4	6\$:	MOVL	#917504, (Q_ICURSOR)+	0968
		8B	7C	000AB	CLRL	(Q_ICURSOR)+	0970	
		04	000AD	7\$:	RET		0974	

; Routine Size: 174 bytes, Routine Base: CODE + 02E9



```

879 0975 1 GLOBAL ROUTINE PARSE_CHAR_NUMBER(PARSE_PARAMETERS_): PARSE_LINKAGE=
880 0976 1
881 0977 1 !++
882 0978 1
883 0979 1 FUNCTIONAL DESCRIPTION:
884 0980 1 This routine parses a characteristic number, making an entry in
885 0981 1 the job controller parameter list.
886 0982 1
887 0983 1 INPUT PARAMETERS:
888 0984 1 Standard parser parameters.
889 0985 1
890 0986 1 IMPLICIT INPUTS:
891 0987 1 NONE
892 0988 1
893 0989 1 OUTPUT PARAMETERS:
894 0990 1 NONE
895 0991 1
896 0992 1 IMPLICIT OUTPUTS:
897 0993 1 NONE
898 0994 1
899 0995 1 ROUTINE VALUE:
900 0996 1 NONE
901 0997 1
902 0998 1 SIDE EFFECTS:
903 0999 1 NONE
904 1000 1
905 1001 1 !--
906 1002 1
907 1003 2 BEGIN
908 1004 2 PARSE_EXTERNAL_REGISTERS; ! Declare external registers
909 1005 2 LOCAL
910 1006 2 STATUS;
911 1007 2
912 1008 2
913 1009 2 CLISGET_VALUE(.Q_DESC, .Q_VALUE_DESC);
914 1010 2 IF
915 1011 2 BEGIN
916 1012 2 IF NOT LIB$CVT_DTB(
917 1013 2 .Q_VALUE_DESC[DSC$W_LENGTH], .Q_VALUE_DESC[DSC$A_POINTER],
918 1014 2 .Q_DCUSOR)
919 1015 2 THEN
920 1016 2 TRUE
921 1017 2 ELSE
922 1018 2 .Q_DCUSOR[0,0,32,0] GTRU 127 ! 0 <= N <= 127
923 1019 2 END
924 1020 2 THEN
925 1021 2 SIGNAL_STOP(.Q_MESSAGE, 2, .Q_VALUE_DESC, .Q_DESC);
926 1022 2
927 1023 2
928 1024 2 Q_ICUSOR[0,0,16,0] = 4;
929 1025 2 Q_ICUSOR[2,0,16,0] = SJCS_CHARACTERISTIC_NUMBER;
930 1026 2 Q_ICUSOR[4,0,32,0] = .Q_DCUSOR;
931 1027 2 Q_ICUSOR[8,0,32,0] = 0;
932 1028 2 Q_ICUSOR = .Q_ICUSOR + 12;
933 1029 2 Q_DCUSOR = .Q_DCUSOR + 4;
934 1030 1 END;
```



			0000 00000	.ENTRY	PARSE CHAR NUMBER, Save nothing	: 0975
			58 DD 00002	PUSHL	Q_VALUE_DESC	: 1009
00000000G	00	04	AC DD 00004	PUSHL	Q_DESC	:
			02 FB 00007	CALLS	#2, CLISGET_VALUE	:
			5A DD 0000E	PUSHL	Q_DCURSOR	: 1014
		04	A8 DD 00010	PUSHL	4Q_VALUE_DESC)	: 1013
	7E		68 3C 00013	MOVZWL	(Q_VALUE_DESC), -(SP)	:
00000000G	00		03 FB 00016	CALLS	#3, LIB\$CVT_DTB	:
	09		50 E9 0001D	BLBC	R0, 1\$	:
0000007F	8F		6A D1 00020	CMPL	(Q_DCURSOR), #127	: 1018
			10 1B 00027	BLEQU	2\$	:
		04	AC DD 00029 1\$:	PUSHL	Q_DESC	: 1021
			58 DD 0002C	PUSHL	Q_VALUE_DESC	:
			02 DD 0002E	PUSHL	#2	:
			59 DD 00030	PUSHL	Q_MESSAGE	:
00000000G	00		04 FB 00032	CALLS	#4, LIB\$STOP	:
	8B 000D0004		8F D0 00039 2\$:	MOVL	#851972, (Q_ICURSOR)+	: 1024
	8B		8A DE 00040	MOVAL	(Q_DCURSOR)+, (Q_ICURSOR)+	: 1026
			8B D4 00043	CLRL	(Q_ICURSOR)+	: 1027
			04 00045	RET		: 1030

; Routine Size: 70 bytes, Routine Base: CODE + 0397



```
936 1031 1 GLOBAL ROUTINE PARSE_COMMA_LIST(PARSE_PARAMETERS_): PARSE_LINKAGE=
937 1032 1
938 1033 1 !++
939 1034 1
940 1035 1 FUNCTIONAL DESCRIPTION:
941 1036 1 This routine parses a qualifier whose value is a comma-separated list of
942 1037 1 arbitrary strings, making an entry in the job controller parameter list.
943 1038 1
944 1039 1 INPUT PARAMETERS:
945 1040 1 Standard parser parameters.
946 1041 1
947 1042 1 IMPLICIT INPUTS:
948 1043 1 NONE
949 1044 1
950 1045 1 OUTPUT PARAMETERS:
951 1046 1 NONE
952 1047 1
953 1048 1 IMPLICIT OUTPUTS:
954 1049 1 NONE
955 1050 1
956 1051 1 ROUTINE VALUE:
957 1052 1 NONE
958 1053 1
959 1054 1 SIDE EFFECTS:
960 1055 1 NONE
961 1056 1
962 1057 1 !--
963 1058 1
964 1059 2 BEGIN
965 1060 2 PARSE_EXTERNAL_REGISTERS; ! Declare external registers
966 1061 2 LOCAL
967 1062 2 FIRST,
968 1063 2 STATUS;
969 1064 2
970 1065 2
971 1066 2 STATUS = CLISPRESNT(.Q_DESC);
972 1067 2 IF .STATUS
973 1068 2 THEN
974 1069 3 BEGIN
975 1070 3 FIRST = TRUE;
976 1071 3 Q_ICURSOR[0,0,16,0] = 0;
977 1072 3 Q_ICURSOR[2,0,16,0] = .Q_P1;
978 1073 3 Q_ICURSOR[4,0,32,0] = .Q_DCURSOR;
979 1074 3 Q_ICURSOR[8,0,32,0] = 0;
980 1075 3 WHILE CLISGET_VALUE(.Q_DESC, .Q_VALUE_DESC) DO
981 1076 4 BEGIN
982 1077 4 IF TESTBITCC(FIRST)
983 1078 4 THEN
984 1079 5 BEGIN
985 1080 5 (.Q_DCURSOR)<0,8> = %C',';
986 1081 5 Q_DCURSOR = .Q_DCURSOR + 1;
987 1082 5 Q_ICURSOR[0,0,16,0] = .Q_ICURSOR[0,0,16,0] + 1;
988 1083 4 END;
989 1084 4 Q_ICURSOR[0,0,16,0] =
990 1085 4 .Q_ICURSOR[0,0,16,0] + .Q_VALUE_DESC[DSCSW_LENGTH];
991 1086 4 Q_DCURSOR = CHSMOVE(
992 1087 4 .Q_VALUE_DESC[DSCSW_LENGTH],
```

```

: 993      1088 4      .Q_VALUE_DESC[DSC$A_POINTER],
: 994      1089      .Q_DCURSOR);
: 995      1090      END;
: 996      1091      Q_ICURSOR = .Q_ICURSOR + 12;
: 997      1092      END;
: 998      1093      IF .STATUS EQL CLIS_NEGATED OR .STATUS EQL CLIS_LOCNEG
: 999      1094      THEN
: 1000      1095      BEGIN
: 1001      1096      Q_ICURSOR[0,0,16,0] = 0;
: 1002      1097      Q_ICURSOR[2,0,16,0] = .Q_P2;
: 1003      1098      Q_ICURSOR[4,0,32,0] = 0;
: 1004      1099      Q_ICURSOR[8,0,32,0] = 0;
: 1005      1100      Q_ICURSOR = .Q_ICURSOR + 12;
: 1006      1101      END;
: 1007      1102 1 END;
```

			00FC 00000	.ENTRY	PARSE_COMMA_LIST, Save R2,R3,R4,R5,R6,R7	: 1031
		04	AC DD 00002	PUSHL	Q_DESC	: 1066
	00000000G	00	01 FB 00005	CALLS	#T, CLISPRESENT	
		57	50 D0 0000C	MOVL	R0, STATUS	
		39	57 E9 0000F	BLBC	STATUS, 4\$	: 1067
		56	01 D0 00012	MOVL	#1, FIRST	: 1070
			6B B4 00015	CLRW	(Q_ICURSOR)	: 1071
	02 AB	08	AC B0 00017	MOVW	Q_P1, 2(Q_ICURSOR)	: 1072
	04 AB		5A D0 0001C	MOVL	Q_DCURSOR, 4(Q_ICURSOR)	: 1073
		08	AB D4 00020	CLRL	8(Q_ICURSOR)	: 1074
			58 DD 00023 1\$:	PUSHL	Q_VALUE_DESC	: 1075
		04	AC DD 00025	PUSHL	Q_DESC	
	00000000G	00	02 FB 00028	CALLS	#2, CLISGET_VALUE	
		16	50 E9 0002F	BLBC	R0, 3\$	
05		56	00 E4 00032	BBSC	#0, FIRST, 2\$	: 1077
		8A	2C 90 00036	MOVB	#44, (Q_DCURSOR)+	: 1080
			6B B6 00039	INCW	(Q_ICURSOR)	: 1082
	6A	04	68 A0 0003B 2\$:	ADDW2	(Q_VALUE_DESC), (Q_ICURSOR)	: 1085
			68 28 0003E	MOVC3	(Q_VALUE_DESC), @4(Q_VALUE_DESC), -	: 1089
					(Q_DCURSOR)	
			53 D0 00043	MOVL	R3, Q_DCURSOR	
			DB 11 00046	BRB	1\$	: 1075
			0C C0 00048 3\$:	ADDL2	#12, Q_ICURSOR	: 1091
	00000000G	8F	57 D1 0004B 4\$:	CMPL	STATUS, #CLIS_NEGATED	: 1093
			09 13 00052	BEQL	5\$	
	00000000G	8F	57 D1 00054	CMPL	STATUS, #CLIS_LOCNEG	
			08 12 0005B	BNEQ	6\$	
			8B B4 0005D 5\$:	CLRW	(Q_ICURSOR)+	: 1096
		8B	AC B0 0005F	MOVW	Q_P2, (Q_ICURSOR)+	: 1097
			8B 7C 00063	CLRW	(Q_ICURSOR)+	: 1098
			04 00065 6\$:	RET		: 1102

; Routine Size: 102 bytes, Routine Base: CODE + 03DD



```
1009 1103 1 GLOBAL ROUTINE PARSE_COPIES(PARSE_PARAMETERS_): PARSE_LINKAGE=
1010 1104 1
1011 1105 1 !++
1012 1106 1
1013 1107 1 FUNCTIONAL DESCRIPTION:
1014 1108 1 This routine parses the /COPIES qualifier, making an entry in
1015 1109 1 the job controller parameter list.
1016 1110 1
1017 1111 1 INPUT PARAMETERS:
1018 1112 1 Standard parser parameters.
1019 1113 1
1020 1114 1 IMPLICIT INPUTS:
1021 1115 1 NONE
1022 1116 1
1023 1117 1 OUTPUT PARAMETERS:
1024 1118 1 NONE
1025 1119 1
1026 1120 1 IMPLICIT OUTPUTS:
1027 1121 1 NONE
1028 1122 1
1029 1123 1 ROUTINE VALUE:
1030 1124 1 NONE
1031 1125 1
1032 1126 1 SIDE EFFECTS:
1033 1127 1 NONE
1034 1128 1
1035 1129 1 !--
1036 1130 1
1037 1131 2 BEGIN
1038 1132 2 PARSE_EXTERNAL_REGISTERS; ! Declare external registers
1039 1133 2
1040 1134 2
1041 1135 2 IF CLISGET_VALUE(.Q_DESC, .Q_VALUE_DESC)
1042 1136 2 THEN
1043 1137 3 BEGIN
1044 1138 3 IF
1045 1139 4 BEGIN
1046 1140 4 IF NOT LIB$CVT_DTB(
1047 1141 4 .Q_VALUE_DESC[DSC$W_LENGTH], .Q_VALUE_DESC[DSC$A_POINTER],
1048 1142 4 .Q_DCUSOR)
1049 1143 4 THEN
1050 1144 4 TRUE
1051 1145 4 ELSE
1052 1146 4 .Q_DCUSOR[0,0,32,0] - 1 GTRU 255 - 1 ! 1 <= N <= 255
1053 1147 4 END
1054 1148 3 THEN
1055 1149 3 SIGNAL_STOP(.Q_MESSAGE, 2, .Q_VALUE_DESC, .Q_DESC);
1056 1150 3
1057 1151 3
1058 1152 3 Q_ICUSOR[0,0,16,0] = 4;
1059 1153 3 Q_ICUSOR[2,0,16,0] = .Q_P1;
1060 1154 3 Q_ICUSOR[4,0,32,0] = .Q_DCUSOR;
1061 1155 3 Q_ICUSOR[8,0,32,0] = 0;
1062 1156 3 Q_ICUSOR = .Q_ICUSOR + 12;
1063 1157 3 Q_DCUSOR = .Q_DCUSOR + 4;
1064 1158 2 END;
1065 1159 1 END;
```

			0000 00000	.ENTRY	PARSE COPIES, Save nothing	: 1103
			58 DD 00002	PUSHL	Q_VALUE_DESC	: 1135
		04	AC DD 00004	PUSHL	Q_DESC	
	00000000G	00	02 FB 00007	CALLS	#2, CLISGET_VALUE	
		3B	50 E9 0000E	BLBC	R0, 3\$	
			5A DD 00011	PUSHL	Q_DCURSOR	: 1142
		04	A8 DD 00013	PUSHL	47Q_VALUE_DESC)	: 1141
			68 3C 00016	MOVZWL	(Q_VALUE_DESC), -(SP)	
	00000000G	7E	03 FB 00019	CALLS	#3, LIB\$CVT_DTB	
		00	50 E9 00020	BLBC	R0, 1\$	
50		0D	01 C3 00023	SUBL3	#1, (Q_DCURSOR), R0	: 1146
	000000FE	6A	50 D1 00027	CMPL	R0, #254	
		8F	10 1B 0002E	BLEQU	2\$	
			04 AC DD 00030 1\$:	PUSHL	Q_DESC	: 1149
			58 DD 00033	PUSHL	Q_VALUE_DESC	
			02 DD 00035	PUSHL	#2	
			59 DD 00037	PUSHL	Q_MESSAGE	
	00000000G	00	04 FB 00039	CALLS	#4, LIB\$STOP	
		8B	04 B0 00040 2\$:	MOVW	#4, (Q_ICURSOR)+	: 1152
		8B	08 AC B0 00043	MOVW	Q_P1, 7Q_ICURSOR)+	: 1153
		8B	8A DE 00047	MOVAL	(Q_DCURSOR)+, (Q_ICURSOR)+	: 1154
			8B D4 0004A	CLRL	(Q_ICURSOR)+	: 1155
			04 0004C 3\$:	RET		: 1159

; Routine Size: 77 bytes, Routine Base: CODE + 0443



```
1067 1160 1 GLOBAL ROUTINE PARSE_CPUTIME(PARSE_PARAMETERS_): PARSE_LINKAGE=
1068 1161 1
1069 1162 1 !++
1070 1163 1
1071 1164 1 FUNCTIONAL DESCRIPTION:
1072 1165 1 This routine parses the /CPUTIME qualifier, making an entry in
1073 1166 1 the job controller parameter list.
1074 1167 1
1075 1168 1 INPUT PARAMETERS:
1076 1169 1 Standard parser parameters.
1077 1170 1
1078 1171 1 IMPLICIT INPUTS:
1079 1172 1 NONE
1080 1173 1
1081 1174 1 OUTPUT PARAMETERS:
1082 1175 1 NONE
1083 1176 1
1084 1177 1 IMPLICIT OUTPUTS:
1085 1178 1 NONE
1086 1179 1
1087 1180 1 ROUTINE VALUE:
1088 1181 1 NONE
1089 1182 1
1090 1183 1 SIDE EFFECTS:
1091 1184 1 NONE
1092 1185 1
1093 1186 1 --
1094 1187 1
1095 1188 2 BEGIN
1096 1189 2 PARSE_EXTERNAL_REGISTERS; ! Declare external registers
1097 1190 2
1098 1191 2
1099 1192 2 IF CLISGET_VALUE(.Q_DESC, .Q_VALUE_DESC)
1100 1193 2 THEN
1101 1194 3 BEGIN
1102 1195 3 LOCAL
1103 1196 3 TIME_VALUE: VECTOR[2],
1104 1197 3 REMAINDER;
1105 1198 3
1106 1199 3
1107 1200 3 IF CALL_TPASE(.Q_VALUE_DESC, NONE_STATES, NONE_KEYS)
1108 1201 3 THEN
1109 1202 4 BEGIN
1110 1203 4 Q_ICURSOR[0,0,16,0] = 0;
1111 1204 4 Q_ICURSOR[2,0,16,0] = .Q_P2;
1112 1205 4 Q_ICURSOR[4,0,32,0] = 0;
1113 1206 4 Q_ICURSOR[8,0,32,0] = 0;
1114 1207 4 Q_ICURSOR = .Q_ICURSOR + 12;
1115 1208 4 END
1116 1209 3 ELSE IF CALL_TPASE(.Q_VALUE_DESC, INFI_STATES, INFI_KEYS)
1117 1210 3 THEN
1118 1211 4 BEGIN
1119 1212 4 Q_ICURSOR[0,0,16,0] = 4;
1120 1213 4 Q_ICURSOR[2,0,16,0] = .Q_P1;
1121 1214 4 Q_ICURSOR[4,0,32,0] = LITERAL_ZERO;
1122 1215 4 Q_ICURSOR[8,0,32,0] = 0;
1123 1216 4 Q_ICURSOR = .Q_ICURSOR + 12;
```

```
1124      1217 4      END
1125      1218 3      ELSE
1126      1219 4      BEGIN
1127      1220 4      IF
1128      1221 5      BEGIN
1129      1222 5      IF NOT LIB$CVT_DTIME(.Q_VALUE_DESC, TIME_VALUE)
1130      1223 5      THEN
1131      1224 5      TRUE
1132      1225 5      ELSE
1133      1226 5      (EDIV(XREF(-200000), TIME_VALUE, .Q_DCUSOR, REMAINDER) AND PSL$M_V) NEQ 0
1134      1227 5      END
1135      1228 4      THEN
1136      1229 4      SIGNAL_STOP(.Q_MESSAGE, 2, .Q_VALUE_DESC, .Q_DESC);
1137      1230 4
1138      1231 4
1139      1232 4      Q_DCUSOR[0,0,32,0] = .Q_DCUSOR[0,0,32,0] ^ 1;
1140      1233 4      IF .REMAINDER NEQ 0 THEN Q_DCUSOR[0,0,32,0] = .Q_DCUSOR[0,0,32,0] + 1;
1141      1234 4      Q_ICUSOR[0,0,16,0] = 4;
1142      1235 4      Q_ICUSOR[2,0,16,0] = .Q_P1;
1143      1236 4      Q_ICUSOR[4,0,32,0] = .Q_DCUSOR;
1144      1237 4      Q_ICUSOR[8,0,32,0] = 0;
1145      1238 4      Q_ICUSOR = .Q_ICUSOR + 12;
1146      1239 4      Q_DCUSOR = .Q_DCUSOR + 4;
1147      1240 3      END;
1148      1241 2      END;
1149      1242 1      END;
```

			000C 00000	.ENTRY	PARSE_CPUTIME, Save R2,R3	: 1160
	53	FB76	CF 9E 00002	MOVAB	CALL_TPARSE, R3	
	5E		08 C2 00007	SUBL2	#8, SP	
		04	58 DD 0000A	PUSHL	Q_VALUE_DESC	: 1192
00000000G	00		AC DD 0000C	PUSHL	Q_DESC	
	01		02 FB 0000F	CALLS	#2, CLISGET_VALUE	
			50 E8 00016	BLBS	R0, 1\$	
			04 00019	RET		
		0000V	CF 9F 0001A 1\$:	PUSHAB	NONE_KEYS	: 1200
		0000V	CF 9F 0001E	PUSHAB	NONE_STATES	
			58 DD 00022	PUSHL	Q_VALUE_DESC	
	63		03 FB 00024	CALLS	#3, CALL_TPARSE	
	0C		50 E9 00027	BLBC	R0, 2\$	
			6B B4 0002A	CLRW	(Q_ICUSOR)	: 1203
02	AB	0C	AC B0 0002C	MOVW	Q_P2, 2(Q_ICUSOR)	: 1204
		04	AB D4 00031	CLRL	4(Q_ICUSOR)	: 1205
			62 11 00034	BRB	7\$	: 1206
		0000V	CF 9F 00036 2\$:	PUSHAB	INFI_KEYS	: 1209
		0000V	CF 9F 0003A	PUSHAB	INFI_STATES	
			58 DD 0003E	PUSHL	Q_VALUE_DESC	
	63		03 FB 00040	CALLS	#3, CALL_TPARSE	
	10		50 E9 00043	BLBC	R0, 3\$	
	6B		04 B0 00046	MOVW	#4, (Q_ICUSOR)	: 1212
02	AB	08	AC B0 00049	MOVW	Q_P1, 2(Q_ICUSOR)	: 1213
04	AB	FB22	CF 9E 0004E	MOVAB	LITERAL_ZERO, 4(Q_ICUSOR)	: 1214
			42 11 00054	BRB	7\$	: 1215



52	00000000G	00	4100	8F	BB	00056	3\$:	PUSHR	#^M<R8,SP>	1222
		0F		02	FB	0005A		CALLS	#2, LIB\$CVT_DTIME	
	6A	6E	FFFCF2C0	50	E9	00061		BLBC	R0, 4\$	
				8F	7B	00064		EDIV	#-200000, TIME_VALUE, (Q_DCURSOR), -	1226
									REMAINDER	
				50	DC	0006D		MOVPSL	R0	
	10	50		01	E1	0006F		BBC	#1, R0, 5\$	
			04	AC	DD	00073	4\$:	PUSHL	Q_DESC	1229
				58	DD	00076		PUSHL	Q_VALUE_DESC	
				02	DD	00078		PUSHL	#2	
				59	DD	0007A		PUSHL	Q_MESSAGE	
	00000000G	00		04	FB	0007C		CALLS	#4, LIB\$STOP	
		6A		02	C4	00083	5\$:	MULL2	#2, (Q_DCURSOR)	1232
				52	D5	00086		TSTL	REMAINDER	1233
				02	13	00088		BEQL	6\$	
				6A	D6	0008A		INCL	(Q_DCURSOR)	
		6B		04	B0	0008C	6\$:	MOVW	#4, (Q_ICURSOR)	1234
	02	AB	08	AC	B0	0008F		MOVW	Q_P1, 2(Q_ICURSOR)	1235
	04	AB		8A	DE	00094		MOVAL	(Q_DCURSOR)+, 4(Q_ICURSOR)	1236
			08	AB	D4	00098	7\$:	CLRL	8(Q_ICURSOR)	1237
		5B		0C	C0	0009B		ADDL2	#12, Q_ICURSOR	1207
				04	0009E			RET		1242

; Routine Size: 159 bytes, Routine Base: CODE + 0490

```
: 1151 1243 1 GLOBAL ROUTINE PARSE_ENTRY(PARSE_PARAMETERS_): PARSE_LINKAGE_VALUE=
: 1152 1244 1
: 1153 1245 1 !++
: 1154 1246 1
: 1155 1247 1 FUNCTIONAL DESCRIPTION:
: 1156 1248 1 This routine parses the /ENTRY qualifier, making an entry in
: 1157 1249 1 the job controller parameter list.
: 1158 1250 1
: 1159 1251 1 INPUT PARAMETERS:
: 1160 1252 1 Standard parser parameters.
: 1161 1253 1
: 1162 1254 1 IMPLICIT INPUTS:
: 1163 1255 1 NONE
: 1164 1256 1
: 1165 1257 1 OUTPUT PARAMETERS:
: 1166 1258 1 NONE
: 1167 1259 1
: 1168 1260 1 IMPLICIT OUTPUTS:
: 1169 1261 1 NONE
: 1170 1262 1
: 1171 1263 1 ROUTINE VALUE:
: 1172 1264 1 As returned from CLISGET_VALUE.
: 1173 1265 1
: 1174 1266 1 SIDE EFFECTS:
: 1175 1267 1 NONE
: 1176 1268 1
: 1177 1269 1 !--
: 1178 1270 1
: 1179 1271 2 BEGIN
: 1180 1272 2 PARSE_EXTERNAL_REGISTERS; ! Declare external registers
: 1181 1273 2 LOCAL
: 1182 1274 2 STATUS;
: 1183 1275 2
: 1184 1276 2
: 1185 1277 2 STATUS = CLISGET_VALUE(.Q_DESC, .Q_VALUE_DESC);
: 1186 1278 2 IF .STATUS
: 1187 1279 2 THEN
: 1188 1280 3 BEGIN
: 1189 1281 3 IF NOT LIB$CVT DTB(
: 1190 1282 3 .Q_VALUE_DESC[DSC$W_LENGTH], .Q_VALUE_DESC[DSC$A_POINTER],
: 1191 1283 3 .Q_DCURSOR)
: 1192 1284 3 THEN
: 1193 1285 3 SIGNAL_STOP(.Q_MESSAGE, 2, .Q_VALUE_DESC, .Q_DESC);
: 1194 1286 3
: 1195 1287 3
: 1196 1288 3 Q_ICURSOR[0,0,16,0] = 4;
: 1197 1289 3 Q_ICURSOR[2,0,16,0] = SJC$ ENTRY_NUMBER;
: 1198 1290 3 Q_ICURSOR[4,0,32,0] = .Q_DCURSOR;
: 1199 1291 3 Q_ICURSOR[8,0,32,0] = 0;
: 1200 1292 3 Q_ICURSOR = .Q_ICURSOR + 12;
: 1201 1293 3 Q_DCURSOR = .Q_DCURSOR + 4;
: 1202 1294 2 END;
: 1203 1295 2
: 1204 1296 2
: 1205 1297 2 .STATUS
: 1206 1298 1 END;
```



			0004 00000	.ENTRY	PARSE_ENTRY, Save R2	: 1243
			58 DD 00002	PUSHL	Q_VALUE_DESC	: 1277
		04	AC DD 00004	PUSHL	Q_DESC	
00000000G	00		02 FB 00007	CALLS	#2, CLISGET_VALUE	
	52		50 D0 0000E	MOVL	R0, STATUS	
	2E		52 E9 00011	BLBC	STATUS, 2\$	: 1278
			5A DD 00014	PUSHL	Q_DCURSOR	: 1283
		04	A8 DD 00016	PUSHL	4(Q_VALUE_DESC)	: 1282
	7E		68 3C 00019	MOVZWL	(Q_VALUE_DESC), -(SP)	
00000000G	00		03 FB 0001C	CALLS	#3, LIB\$CVT_DTB	
	10		50 E8 00023	BLBS	R0, 1\$	
		04	AC DD 00026	PUSHL	Q_DESC	: 1285
			58 DD 00029	PUSHL	Q_VALUE_DESC	
			02 DD 0002B	PUSHL	#2	
			59 DD 0002D	PUSHL	Q_MESSAGE	
00000000G	00		04 FB 0002F	CALLS	#4, LIB\$STOP	
	8B	001E0004	8F D0 00036 1\$:	MOVL	#1966084, (Q_ICURSOR)+	: 1288
	8B		8A DE 0003D	MOVAL	(Q_DCURSOR)+, (Q_ICURSOR)+	: 1290
			8B D4 00040	CLRL	(Q_ICURSOR)+	: 1291
	50		52 D0 00042 2\$:	MOVL	STATUS, R0	: 1298
			04 00045	RET		

; Routine Size: 70 bytes, Routine Base: CODE + 052F



```
: 1208      1299 1 GLOBAL ROUTINE PARSE_EXTEND_QUANTITY(PARSE_PARAMETERS_): PARSE_LINKAGE=
: 1209      1300 1
: 1210      1301 1 !++
: 1211      1302 1
: 1212      1303 1 FUNCTIONAL DESCRIPTION:
: 1213      1304 1     This routine parses the /EXTEND_QUANTITY qualifier, making an entry in
: 1214      1305 1     the job controller parameter list.
: 1215      1306 1
: 1216      1307 1 INPUT PARAMETERS:
: 1217      1308 1     Standard parser parameters.
: 1218      1309 1
: 1219      1310 1 IMPLICIT INPUTS:
: 1220      1311 1     NONE
: 1221      1312 1
: 1222      1313 1 OUTPUT PARAMETERS:
: 1223      1314 1     NONE
: 1224      1315 1
: 1225      1316 1 IMPLICIT OUTPUTS:
: 1226      1317 1     NONE
: 1227      1318 1
: 1228      1319 1 ROUTINE VALUE:
: 1229      1320 1     NONE
: 1230      1321 1
: 1231      1322 1 SIDE EFFECTS:
: 1232      1323 1     NONE
: 1233      1324 1
: 1234      1325 1 --
: 1235      1326 1
: 1236      1327 2 BEGIN
: 1237      1328 2 PARSE_EXTERNAL_REGISTERS;      ! Declare external registers
: 1238      1329 2
: 1239      1330 2
: 1240      1331 2 IF CLISGET_VALUE(.Q_DESC, .Q_VALUE_DESC)
: 1241      1332 2 THEN
: 1242      1333 3     BEGIN
: 1243      1334 3     IF
: 1244      1335 4         BEGIN
: 1245      1336 4         IF NOT LIB$CVT_DTB(
: 1246      1337 4             .Q_VALUE_DESC[DSC$W_LENGTH], .Q_VALUE_DESC[DSC$A_POINTER],
: 1247      1338 4             .Q_DCURSOR)
: 1248      1339 4         THEN
: 1249      1340 4             TRUE
: 1250      1341 4         ELSE
: 1251      1342 4             .Q_DCURSOR[0,0,32,0] GTRU 65535      ! 0 <= N <= 65535
: 1252      1343 4         END
: 1253      1344 3     THEN
: 1254      1345 3         SIGNAL_STOP(.Q_MESSAGE, 2, .Q_VALUE_DESC, .Q_DESC);
: 1255      1346 3
: 1256      1347 3
: 1257      1348 3     Q_ICURSOR[0,0,16,0] = 4;
: 1258      1349 3     Q_ICURSOR[2,0,16,0] = SJC$ EXTEND_QUANTITY;
: 1259      1350 3     Q_ICURSOR[4,0,32,0] = .Q_DCURSOR;
: 1260      1351 3     Q_ICURSOR[8,0,32,0] = 0;
: 1261      1352 3     Q_ICURSOR = .Q_ICURSOR + 12;
: 1262      1353 3     Q_DCURSOR = .Q_DCURSOR + 4;
: 1263      1354 2     END;
: 1264      1355 1 END;
```



			0000 00000	.ENTRY	PARSE_EXTEND_QUANTITY, Save nothing	: 1299
			58 DD 00002	PUSHL	Q_VALUE_DESC	: 1331
00000000G	00	04	AC DD 00004	PUSHL	Q_DESC	
	37		02 FB 00007	CALLS	#2, CLISGET_VALUE	
			50 E9 0000E	BLBC	R0, 3\$	
			5A DD 00011	PUSHL	Q_DCURSOR	: 1338
		04	A8 DD 00013	PUSHL	47Q VALUE_DESC)	: 1337
	7E		68 3C 00016	MOVZWL	(Q_VALUE_DESC), -(SP)	
00000000G	00		03 FB 00019	CALLS	#3, LIB\$CVT_DTB	
	09		50 E9 00020	BLBC	R0, 1\$	
0000FFFF	8F		6A D1 00023	CMPL	(Q_DCURSOR), #65535	: 1342
			10 1B 0002A	BLEQU	2\$	
		04	AC DD 0002C 1\$:	PUSHL	Q_DESC	: 1345
			58 DD 0002F	PUSHL	Q_VALUE_DESC	
			02 DD 00031	PUSHL	#2	
			59 DD 00033	PUSHL	Q_MESSAGE	
00000000G	00		04 FB 00035	CALLS	#4, LIB\$STOP	
	8B 00A10004		8F D0 0003C 2\$:	MOVL	#10551300, (Q_ICURSOR)+	: 1348
	8B		8A DE 00043	MOVAL	(Q_DCURSOR)+, (Q_ICURSOR)+	: 1350
			8B D4 00046	CLRL	(Q_ICURSOR)+	: 1351
			04 00048 3\$:	RET		: 1355

; Routine Size: 73 bytes, Routine Base: CODE + 0575

```
1266 1356 1 GLOBAL ROUTINE PARSE_FILENAME(PARSE_PARAMETERS_): PARSE_LINKAGE=
1267 1357 1
1268 1358 1 ++
1269 1359 1
1270 1360 1 FUNCTIONAL DESCRIPTION:
1271 1361 1 This routine parses a qualifier whose value is a file name, making an
1272 1362 1 entry in the job controller parameter list.
1273 1363 1
1274 1364 1 INPUT PARAMETERS:
1275 1365 1 Standard parser parameters.
1276 1366 1
1277 1367 1 IMPLICIT INPUTS:
1278 1368 1 NONE
1279 1369 1
1280 1370 1 OUTPUT PARAMETERS:
1281 1371 1 NONE
1282 1372 1
1283 1373 1 IMPLICIT OUTPUTS:
1284 1374 1 NONE
1285 1375 1
1286 1376 1 ROUTINE VALUE:
1287 1377 1 NONE
1288 1378 1
1289 1379 1 SIDE EFFECTS:
1290 1380 1 NONE
1291 1381 1
1292 1382 1 --
1293 1383 1
1294 1384 2 BEGIN
1295 1385 2 PARSE_EXTERNAL_REGISTERS; ! Declare external registers
1296 1386 2 LOCAL
1297 1387 2 STATUS;
1298 1388 2
1299 1389 2
1300 1390 2 STATUS = CLISPRESENT(.Q_DESC);
1301 1391 2 IF .STATUS
1302 1392 2 THEN
1303 1393 3 BEGIN
1304 1394 3 CLISGET VALUE(.Q_DESC, .Q_VALUE_DESC);
1305 1395 3 IF .Q_VALUE_DESC[DSC$W_LENGTH] = 1 GTRU 39 - 1 ! 1 <= N <= 39
1306 1396 3 THEN
1307 1397 3 SIGNAL_STOP(.Q_MESSAGE, 2, .Q_VALUE_DESC, .Q_DESC);
1308 1398 3
1309 1399 3
1310 1400 3 Q_ICURSOR[0,0,16,0] = .Q_VALUE_DESC[DSC$W_LENGTH];
1311 1401 3 Q_ICURSOR[2,0,16,0] = .Q_P1;
1312 1402 3 Q_ICURSOR[4,0,32,0] = .Q_DCURSOR;
1313 1403 3 Q_ICURSOR[8,0,32,0] = 0;
1314 1404 3 Q_ICURSOR = .Q_ICURSOR + 12;
1315 1405 3 Q_DCURSOR = CHSMOVE(
1316 1406 3 .Q_VALUE_DESC[DSC$W_LENGTH],
1317 1407 3 .Q_VALUE_DESC[DSC$A_POINTER],
1318 1408 3 .Q_DCURSOR);
1319 1409 2 END;
1320 1410 2 IF .STATUS EQL CLIS_NEGATED OR .STATUS EQL CLIS_LOCNEG
1321 1411 2 THEN
1322 1412 3 BEGIN
```



```
: 1323      1413 3      Q_ICURSOR[0,0,16,0] = 0;  
: 1324      1414 3      Q_ICURSOR[2,0,16,0] = .Q_P2;  
: 1325      1415 3      Q_ICURSOR[4,0,32,0] = 0;  
: 1326      1416 3      Q_ICURSOR[8,0,32,0] = 0;  
: 1327      1417 3      Q_ICURSOR = .Q_ICURSOR + 12;  
: 1328      1418 2      END;  
: 1329      1419 1 END;
```

			007C 00000	.ENTRY	PARSE_FILENAME, Save R2,R3,R4,R5,R6	: 1356
		04	AC DD 00002	PUSHL	Q_DESC	: 1390
00000000G	00		01 FB 00005	CALLS	#T, CLISPRESENT	
	56		50 D0 0000C	MOVL	R0, STATUS	
	3A		56 E9 0000F	BLBC	STATUS, 2\$	: 1391
		04	58 DD 00012	PUSHL	Q_VALUE_DESC	: 1394
00000000G	00		AC DD 00014	PUSHL	Q_DESC	
	50		02 FB 00017	CALLS	#2, CLISGET_VALUE	
			68 3C 0001E	MOVZWL	(Q_VALUE_DESC), R0	: 1395
	26		50 D7 00021	DECL	R0	
			50 D1 00023	CML	R0, #38	
		04	10 1B 00026	BLEQU	1\$	
			AC DD 00028	PUSHL	Q_DESC	: 1397
			58 DD 0002B	PUSHL	Q_VALUE_DESC	
			02 DD 0002D	PUSHL	#2	
00000000G	00		59 DD 0002F	PUSHL	Q_MESSAGE	
	8B		04 FB 00031	CALLS	#Z, LIB\$STOP	
	8B		68 B0 00038 1\$:	MOVW	(Q_VALUE_DESC), (Q_ICURSOR)+	: 1400
	8B	08	AC B0 0003B	MOVW	Q_P1, (Q_ICURSOR)+	: 1401
			5A D0 0003F	MOVL	Q_DCURSOR, (Q_ICURSOR)+	: 1402
6A	04	BB	8B D4 00042	CLRL	(Q_ICURSOR)+	: 1403
			68 28 00044	MOVCL	(Q_VALUE_DESC), @4(Q_VALUE_DESC), -	: 1408
					(Q_DCURSOR)	
00000000G	5A		53 D0 00049	MOVL	R3, Q_DCURSOR	
	8F		56 D1 0004C 2\$:	CML	STATUS, #CLIS_NEGATED	: 1410
00000000G	8F		09 13 00053	BEQL	3\$	
			56 D1 00055	CML	STATUS, #CLIS_LOCNEG	
			08 12 0005C	BNEQ	4\$	
	8B		8B B4 0005E 3\$:	CLRW	(Q_ICURSOR)+	: 1413
		0C	AC B0 00060	MOVW	Q_P2, (Q_ICURSOR)+	: 1414
			8B 7C 00064	CLRW	(Q_ICURSOR)+	: 1415
			04 00066 4\$:	RET		: 1419

; Routine Size: 103 bytes, Routine Base: CODE + 05BE

```
: 1331      1420 1 GLOBAL ROUTINE PARSE_FORM(PARSE_PARAMETERS_): PARSE_LINKAGE=
: 1332      1421 1
: 1333      1422 1 !++
: 1334      1423 1
: 1335      1424 1 FUNCTIONAL DESCRIPTION:
: 1336      1425 1     This routine parses the /FORM qualifier, making an entry in
: 1337      1426 1     the job controller parameter list.
: 1338      1427 1
: 1339      1428 1 INPUT PARAMETERS:
: 1340      1429 1     Standard parser parameters.
: 1341      1430 1
: 1342      1431 1 IMPLICIT INPUTS:
: 1343      1432 1     NONE
: 1344      1433 1
: 1345      1434 1 OUTPUT PARAMETERS:
: 1346      1435 1     NONE
: 1347      1436 1
: 1348      1437 1 IMPLICIT OUTPUTS:
: 1349      1438 1     NONE
: 1350      1439 1
: 1351      1440 1 ROUTINE VALUE:
: 1352      1441 1     NONE
: 1353      1442 1
: 1354      1443 1 SIDE EFFECTS:
: 1355      1444 1     NONE
: 1356      1445 1
: 1357      1446 1 !--
: 1358      1447 1
: 1359      1448 2 BEGIN
: 1360      1449 2 PARSE_EXTERNAL_REGISTERS;      ! Declare external registers
: 1361      1450 2
: 1362      1451 2
: 1363      1452 2 IF CLISGET_VALUE(.Q_DESC, .Q_VALUE_DESC)
: 1364      1453 2 THEN
: 1365      1454 3     BEGIN
: 1366      1455 3         IF LIB$CVT_DTB(
: 1367      1456 3             .Q_VALUE_DESC[DSC$W_LENGTH], .Q_VALUE_DESC[DSC$A_POINTER],
: 1368      1457 3             .Q_DCUSOR)
: 1369      1458 3         THEN
: 1370      1459 4             BEGIN
: 1371      1460 4                 Q_ICUSOR[0,0,16,0] = 4;
: 1372      1461 4                 Q_ICUSOR[2,0,16,0] = SJC$ FORM NUMBER;
: 1373      1462 4                 Q_ICUSOR[4,0,32,0] = .Q_DCUSOR;
: 1374      1463 4                 Q_ICUSOR[8,0,32,0] = 0;
: 1375      1464 4                 Q_ICUSOR = .Q_ICUSOR + 12;
: 1376      1465 4                 Q_DCUSOR = .Q_DCUSOR + 4;
: 1377      1466 4             END
: 1378      1467 3         ELSE IF CALL_TPASE(.Q_VALUE_DESC, SYMB_STATES, SYMB_KEYS)
: 1379      1468 3         THEN
: 1380      1469 4             BEGIN
: 1381      1470 4                 Q_ICUSOR[0,0,16,0] = .Q_VALUE_DESC[DSC$W_LENGTH];
: 1382      1471 4                 Q_ICUSOR[2,0,16,0] = SJC$ FORM NAME;
: 1383      1472 4                 Q_ICUSOR[4,0,32,0] = .Q_DCUSOR;
: 1384      1473 4                 Q_ICUSOR[8,0,32,0] = 0;
: 1385      1474 4                 Q_ICUSOR = .Q_ICUSOR + 12;
: 1386      1475 4                 Q_DCUSOR = CH$MOVE(
: 1387      1476 4                     .Q_VALUE_DESC[DSC$W_LENGTH],
```



```
: 1388      1477 4      .Q_VALUE_DESC[DSC$A_POINTER],  
: 1389      1478 4      .Q_DCUSOR);  
: 1390      1479 4      END  
: 1391      1480 3      ELSE  
: 1392      1481 3      SIGNAL_STOP(.Q_MESSAGE, 2, .Q_VALUE_DESC, .Q_DESC);  
: 1393      1482 2      END;  
: 1394      1483 1 END;
```

			003C 00000	.ENTRY	PARSE FORM, Save R2,R3,R4,R5	: 1420
			58 DD 00002	PUSHL	Q_VALUE_DESC	: 1452
		04	AC DD 00004	PUSHL	Q_DESC	
00000000G	00		02 FB 00007	CALLS	#2, CLISGET_VALUE	
	55		50 E9 0000E	BLBC	R0, 3\$	
		04	5A DD 00011	PUSHL	Q_DCUSOR	: 1457
			A8 DD 00013	PUSHL	4(Q_VALUE_DESC)	: 1456
	7E		68 3C 00016	MOVZWL	(Q_VALUE_DESC), -(SP)	
00000000G	00		03 FB 00019	CALLS	#3, LIB\$CVT_DTB	
	0D		50 E9 00020	BLBC	R0, 1\$	
	8B	00370004	8F D0 00023	MOVL	#3604484, (Q_ICUSOR)+	: 1460
	8B		8A DE 0002A	MOVAL	(Q_DCUSOR)+, (Q_ICUSOR)+	: 1462
			8B D4 0002D	CLRL	(Q_ICUSOR)+	: 1463
			04 0002F	RET		: 1455
		0000V	CF 9F 00030	PUSHAB	SYMB_KEYS	: 1467
		0000V	CF 9F 00034	PUSHAB	SYMB_STATES	
			58 DD 00038	PUSHL	Q_VALUE_DESC	
F9A8	CF		03 FB 0003A	CALLS	#3, CALC_TPARSE	
	14		50 E9 0003F	BLBC	R0, 2\$	
	8B		68 B0 00042	MOVW	(Q_VALUE_DESC), (Q_ICUSOR)+	: 1470
	8B		36 B0 00045	MOVW	#54, (Q_ICUSOR)+	: 1471
	8B		5A D0 00048	MOVL	Q_DCUSOR, (Q_ICUSOR)+	: 1472
			8B D4 0004B	CLRL	(Q_ICUSOR)+	: 1473
6A	04	B8	68 28 0004D	MOVC3	(Q_VALUE_DESC), @4(Q_VALUE_DESC), -	: 1478
					(Q_DCUSOR)	
		5A	53 D0 00052	MOVL	R3, Q_DCUSOR	
			04 00055	RET		: 1467
		04	AC DD 00056	PUSHL	Q_DESC	: 1481
			58 DD 00059	PUSHL	Q_VALUE_DESC	
			02 DD 0005B	PUSHL	#2	
			59 DD 0005D	PUSHL	Q_MESSAGE	
00000000G	00		04 FB 0005F	CALLS	#4, LIB\$STOP	
			04 00066	RET		: 1483

; Routine Size: 103 bytes, Routine Base: CODE + 0625

```
1396 1484 1 GLOBAL ROUTINE PARSE_FORWARD(PARSE_PARAMETERS_): PARSE_LINKAGE=
1397 1485 1
1398 1486 1 !++
1399 1487 1
1400 1488 1 FUNCTIONAL DESCRIPTION:
1401 1489 1 This routine parses the /FORWARD qualifier, making an entry in the
1402 1490 1 job controller parameter list.
1403 1491 1
1404 1492 1 INPUT PARAMETERS:
1405 1493 1 Standard parser parameters.
1406 1494 1
1407 1495 1 IMPLICIT INPUTS:
1408 1496 1 NONE
1409 1497 1
1410 1498 1 OUTPUT PARAMETERS:
1411 1499 1 NONE
1412 1500 1
1413 1501 1 IMPLICIT OUTPUTS:
1414 1502 1 NONE
1415 1503 1
1416 1504 1 ROUTINE VALUE:
1417 1505 1 NONE
1418 1506 1
1419 1507 1 SIDE EFFECTS:
1420 1508 1 NONE
1421 1509 1
1422 1510 1 !--
1423 1511 1
1424 1512 2 BEGIN
1425 1513 2 PARSE_EXTERNAL_REGISTERS; ! Declare external registers
1426 1514 2
1427 1515 2
1428 1516 2 IF CL$PRESENT(.Q_DESC)
1429 1517 2 THEN
1430 1518 3 BEGIN
1431 1519 3 IF CL$GET_VALUE(.Q_DESC, .Q_VALUE_DESC)
1432 1520 3 THEN
1433 1521 4 BEGIN
1434 1522 4 IF
1435 1523 5 BEGIN
1436 1524 5 IF NOT LIB$CVT_DTB(
1437 1525 5 .Q_VALUE_DESC[DSC$W_LENGTH], .Q_VALUE_DESC[DSC$A_POINTER],
1438 1526 5 .Q_DCURSOR)
1439 1527 5 THEN
1440 1528 5 TRUE
1441 1529 5 ELSE
1442 1530 5 .Q_DCURSOR[0,0,32,0] EQL 0
1443 1531 5 END
1444 1532 4 THEN
1445 1533 4 SIGNAL_STOP(.Q_MESSAGE, 2, .Q_VALUE_DESC, .Q_DESC);
1446 1534 4
1447 1535 4
1448 1536 4 Q_ICURSOR[0,0,16,0] = 4;
1449 1537 4 Q_ICURSOR[2,0,16,0] = SJC$ RELATIVE_PAGE;
1450 1538 4 Q_ICURSOR[4,0,32,0] = .Q_DCURSOR;
1451 1539 4 Q_ICURSOR[8,0,32,0] = 0;
1452 1540 4 Q_ICURSOR = .Q_ICURSOR + 12;
```



```
: 1453      1541  4      Q_DCURLOR = .Q_DCURLOR + 4;
: 1454      1542  4      END
: 1455      1543  3      ELSE
: 1456      1544  4      BEGIN
: 1457      1545  4      Q_ICURLOR[0,0,16,0] = 4;
: 1458      1546  4      Q_ICURLOR[2,0,16,0] = SJCS_RELATIVE_PAGE;
: 1459      1547  4      Q_ICURLOR[4,0,32,0] = LITERAL_ONE;
: 1460      1548  4      Q_ICURLOR[8,0,32,0] = 0;
: 1461      1549  4      Q_ICURLOR = .Q_ICURLOR + 12;
: 1462      1550  3      END;
: 1463      1551  2      END;
: 1464      1552  1      END;
```

			0000 00000	.ENTRY	PARSE_FORWARD, Save nothing	: 1484
		04	AC DD 00002	PUSHL	Q_DESC	: 1516
00000000G	00		01 FB 00005	CALLS	#T, CLISPRESENT	
	55		50 E9 0000C	BLBC	R0, 5\$	
		04	58 DD 0000F	PUSHL	Q_VALUE_DESC	: 1519
00000000G	00		AC DD 00011	PUSHL	Q_DESC	
	33		02 FB 00014	CALLS	#2, CLISGET_VALUE	
			50 E9 0001B	BLBC	R0, 3\$	
		04	5A DD 0001E	PUSHL	Q_DCURLOR	: 1526
			A8 DD 00020	PUSHL	4(Q_VALUE_DESC)	: 1525
00000000G	7E		68 3C 00023	MOVZWL	(Q_VALUE_DESC), -(SP)	
	00		03 FB 00026	CALLS	#3, LIB\$CVT_DTB	
	04		50 E9 0002D	BLBC	R0, 1\$	
			6A D5 00030	TSTL	(Q_DCURLOR)	: 1530
		04	10 12 00032	BNEQ	2\$	
			AC DD 00034 1\$:	PUSHL	Q_DESC	: 1533
			58 DD 00037	PUSHL	Q_VALUE_DESC	
			02 DD 00039	PUSHL	#2	
00000000G	00		59 DD 0003B	PUSHL	Q_MESSAGE	
	6B 00880004		04 FB 0003D	CALLS	#4, LIB\$STOP	
	04 AB		8F D0 00044 2\$:	MOVL	#8912900, (Q_ICURLOR)	: 1536
			8A DE 0004B	MOVAL	(Q_DCURLOR)+, 4(Q_ICURLOR)	: 1538
			0D 11 0004F	BRB	4\$	: 1539
	6B 00880004		8F D0 00051 3\$:	MOVL	#8912900, (Q_ICURLOR)	: 1545
	04 AB F920		CF 9E 00058	MOVAB	LITERAL ONE, -4(Q_ICURLOR)	: 1547
		08	AB D4 0005E 4\$:	CLRL	8(Q_ICURLOR)	: 1548
	5B		0C C0 00061	ADDL2	#12, Q_ICURLOR	: 1540
			04 00064 5\$:	RET		: 1552

; Routine Size: 101 bytes, Routine Base: CODE + 068C

```
1466 1553 1 GLOBAL ROUTINE PARSE_GENERIC(PARSE_PARAMETERS_): PARSE_LINKAGE=
1467 1554 1
1468 1555 1 ++
1469 1556 1
1470 1557 1 FUNCTIONAL DESCRIPTION:
1471 1558 1 This routine parses the /GENERIC qualifier, making an entry in
1472 1559 1 the job controller parameter list.
1473 1560 1
1474 1561 1 INPUT PARAMETERS:
1475 1562 1 Standard parser parameters.
1476 1563 1
1477 1564 1 IMPLICIT INPUTS:
1478 1565 1 NONE
1479 1566 1
1480 1567 1 OUTPUT PARAMETERS:
1481 1568 1 NONE
1482 1569 1
1483 1570 1 IMPLICIT OUTPUTS:
1484 1571 1 NONE
1485 1572 1
1486 1573 1 ROUTINE VALUE:
1487 1574 1 NONE
1488 1575 1
1489 1576 1 SIDE EFFECTS:
1490 1577 1 NONE
1491 1578 1
1492 1579 1 --
1493 1580 1
1494 1581 2 BEGIN
1495 1582 2 PARSE_EXTERNAL_REGISTERS; ! Declare external registers
1496 1583 2 LOCAL
1497 1584 2 STATUS;
1498 1585 2
1499 1586 2
1500 1587 2 STATUS = CLISPRESNT(.Q_DESC);
1501 1588 2 IF .STATUS
1502 1589 2 THEN
1503 1590 2 BEGIN
1504 1591 2 Q_ICURSOR[0,0,16,0] = 0;
1505 1592 2 Q_ICURSOR[2,0,16,0] = SJC$_GENERIC_QUEUE;
1506 1593 2 Q_ICURSOR[4,0,32,0] = 0;
1507 1594 2 Q_ICURSOR[8,0,32,0] = 0;
1508 1595 2 Q_ICURSOR = .Q_ICURSOR + 12;
1509 1596 2
1510 1597 2
1511 1598 2 WHILE CLISGET_VALUE(.Q_DESC, .Q_VALUE_DESC) DO
1512 1599 2 BEGIN
1513 1600 2 Q_ICURSOR[0,0,16,0] = .Q_VALUE_DESC[DSC$_LENGTH];
1514 1601 2 Q_ICURSOR[2,0,16,0] = SJC$_GENERIC_TARGET;
1515 1602 2 Q_ICURSOR[4,0,32,0] = .Q_DCURSOR;
1516 1603 2 Q_ICURSOR[8,0,32,0] = 0;
1517 1604 2 Q_ICURSOR = .Q_ICURSOR + 12;
1518 1605 2 Q_DCURSOR = CH$MOVE(
1519 1606 2 .Q_VALUE_DESC[DSC$_LENGTH],
1520 1607 2 .Q_VALUE_DESC[DSC$_A_POINTER],
1521 1608 2 .Q_DCURSOR);
1522 1609 2 END;
```



```
: 1523      1610 2      END;
: 1524      1611 2
: 1525      1612 2
: 1526      1613 2 IF .STATUS EQL CLIS_NEGATED OR .STATUS EQL CLIS_LOCNEG
: 1527      1614 2 THEN
: 1528      1615 2 BEGIN
: 1529      1616 2 Q_ICUSOR[0,0,16,0] = 0;
: 1530      1617 2 Q_ICUSOR[2,0,16,0] = SJCS_NO_GENERIC_QUEUE;
: 1531      1618 2 Q_ICUSOR[4,0,32,0] = 0;
: 1532      1619 2 Q_ICUSOR[8,0,32,0] = 0;
: 1533      1620 2 Q_ICUSOR = .Q_ICUSOR + 12;
: 1534      1621 2 END;
: 1535      1622 1 END;
```

			007C 00000	.ENTRY	PARSE GENERIC, Save R2,R3,R4,R5,R6	: 1553
		04	AC DD 00002	PUSHL	Q_DESC	: 1587
00000000G	00		01 FB 00005	CALLS	#T, CLISPRESENT	
	56		50 D0 0000C	MOVL	R0, STATUS	
	2E		56 E9 0000F	BLBC	STATUS, 2\$	: 1588
	8B	00420000	8F D0 00012	MOVL	#4325376, (Q_ICUSOR)+	: 1591
			8B 7C 00019	CLRQ	(Q_ICUSOR)+	: 1593
			58 DD 0001B 1\$:	PUSHL	Q_VALUE_DESC	: 1598
		04	AC DD 0001D	PUSHL	Q_DESC	
00000000G	00		02 FB 00020	CALLS	#2, CLISGET_VALUE	
	16		50 E9 00027	BLBC	R0, 2\$	
	8B		68 B0 0002A	MOVW	(Q_VALUE_DESC), (Q_ICUSOR)+	: 1600
	8B	46	8F 9B 0002D	MOVZBW	#70, (Q_ICUSOR)+	: 1601
	8B		5A D0 00031	MOVL	Q_DCUSOR, (Q_ICUSOR)+	: 1602
			8B D4 00034	CLRL	(Q_ICUSOR)+	: 1603
6A	04	B8	68 28 00036	MOVCL	(Q_VALUE_DESC), @4(Q_VALUE_DESC), -	: 1608
					(Q_DCUSOR)	
	5A		53 D0 0003B	MOVL	R3, Q_DCUSOR	
			DB 11 0003E	BRB	1\$	: 1598
00000000G	8F		56 D1 00040 2\$:	CMPL	STATUS, #CLIS_NEGATED	: 1613
			09 13 00047	BEQL	3\$	
00000000G	8F		56 D1 00049	CMPL	STATUS, #CLIS_LOCNEG	
			09 12 00050	BNEQ	4\$	
	8B	00430000	8F D0 00052 3\$:	MOVL	#4390912, (Q_ICUSOR)+	: 1616
			8B 7C 00059	CLRQ	(Q_ICUSOR)+	: 1618
			04 0005B 4\$:	RET		: 1622

; Routine Size: 92 bytes, Routine Base: CODE + 06F1

```
: 1537 1623 1 GLOBAL ROUTINE PARSE_JOB_LIMIT(PARSE_PARAMETERS_): PARSE_LINKAGE=
: 1538 1624 1
: 1539 1625 1 ++
: 1540 1626 1
: 1541 1627 1 FUNCTIONAL DESCRIPTION:
: 1542 1628 1 This routine parses the /JOB_LIMIT qualifier, making an entry in
: 1543 1629 1 the job controller parameter list.
: 1544 1630 1
: 1545 1631 1 INPUT PARAMETERS:
: 1546 1632 1 Standard parser parameters.
: 1547 1633 1
: 1548 1634 1 IMPLICIT INPUTS:
: 1549 1635 1 NONE
: 1550 1636 1
: 1551 1637 1 OUTPUT PARAMETERS:
: 1552 1638 1 NONE
: 1553 1639 1
: 1554 1640 1 IMPLICIT OUTPUTS:
: 1555 1641 1 NONE
: 1556 1642 1
: 1557 1643 1 ROUTINE VALUE:
: 1558 1644 1 NONE
: 1559 1645 1
: 1560 1646 1 SIDE EFFECTS:
: 1561 1647 1 NONE
: 1562 1648 1
: 1563 1649 1 --
: 1564 1650 1
: 1565 1651 2 BEGIN
: 1566 1652 2 PARSE_EXTERNAL_REGISTERS; ! Declare external registers
: 1567 1653 2
: 1568 1654 2
: 1569 1655 2 IF CLISGET_VALUE(.Q_DESC, .Q_VALUE_DESC)
: 1570 1656 2 THEN
: 1571 1657 3 BEGIN
: 1572 1658 3 IF
: 1573 1659 4 BEGIN
: 1574 1660 4 IF NOT LIB$CVT_DTB(
: 1575 1661 4 .Q_VALUE_DESC[DSC$W_LENGTH], .Q_VALUE_DESC[DSC$A_POINTER],
: 1576 1662 4 .Q_DCURSOR)
: 1577 1663 4 THEN
: 1578 1664 4 TRUE
: 1579 1665 4 ELSE
: 1580 1666 4 .Q_DCURSOR[0,0,32,0] - 1 GTRU 255 - 1 ! 1 <= N <= 255
: 1581 1667 4 END
: 1582 1668 3 THEN
: 1583 1669 3 SIGNAL_STOP(.Q_MESSAGE, 2, .Q_VALUE_DESC, .Q_DESC);
: 1584 1670 3
: 1585 1671 3
: 1586 1672 3 Q_ICURSOR[0,0,16,0] = 4;
: 1587 1673 3 Q_ICURSOR[2,0,16,0] = SJC$ JOB_LIMIT;
: 1588 1674 3 Q_ICURSOR[4,0,32,0] = .Q_DCURSOR;
: 1589 1675 3 Q_ICURSOR[8,0,32,0] = 0;
: 1590 1676 3 Q_ICURSOR = .Q_ICURSOR + 12;
: 1591 1677 3 Q_DCURSOR = .Q_DCURSOR + 4;
: 1592 1678 2 END;
: 1593 1679 1 END;
```



			0000	00000		.ENTRY	PARSE JOB LIMIT, Save nothing	:	1623
			58	DD	00002	PUSHL	Q_VALUE_DESC	:	1655
		04	AC	DD	00004	PUSHL	Q_DESC	:	
	00000000G	00	02	FB	00007	CALLS	#2, CLISGET_VALUE	:	
		3B	50	E9	0000E	BLBC	R0, 3\$	:	
			5A	DD	00011	PUSHL	Q_DCURSOR	:	1662
		04	A8	DD	00013	PUSHL	4(Q_VALUE_DESC)	:	1661
			68	3C	00016	MOVZWL	(Q_VALUE_DESC), -(SP)	:	
	00000000G	7E	03	FB	00019	CALLS	#3, LIB\$CVT_DTB	:	
		00	50	E9	00020	BLBC	R0, 1\$	:	
		0D	01	C3	00023	SUBL3	#1, (Q_DCURSOR), R0	:	1666
50	000000FE	6A	50	D1	00027	CMPL	R0, #254	:	
		8F	10	1B	0002E	BLEQU	2\$	:	
			04	AC	DD	00030	1\$: PUSHL	:	1669
				58	DD	00033	Q_DESC	:	
				02	DD	00035	Q_VALUE_DESC	:	
				59	DD	00037	#2	:	
	00000000G	00	04	FB	00039	PUSHL	Q_MESSAGE	:	
		8B	04	FB	00039	CALLS	#2, LIB\$STOP	:	
		8B	8F	D0	00040	2\$: MOVL	#5111812, (Q_ICURSOR)+	:	1672
			8A	DE	00047	MOVAL	(Q_DCURSOR)+, (Q_ICURSOR)+	:	1674
			8B	D4	0004A	CLRL	(Q_ICURSOR)+	:	1675
			04	0004C	3\$: RET			:	1679

; Routine Size: 77 bytes, Routine Base: CODE + 074D

```
: 1595 1680 1 GLOBAL ROUTINE PARSE_LOG_FILE(PARSE_PARAMETERS_): PARSE_LINKAGE_VALUE=
: 1596 1681 1
: 1597 1682 1 !++
: 1598 1683 1
: 1599 1684 1 FUNCTIONAL DESCRIPTION:
: 1600 1685 1 This routine parses the /LOG_FILE qualifier, making an entry
: 1601 1686 1 in the job controller parameter list.
: 1602 1687 1
: 1603 1688 1 INPUT PARAMETERS:
: 1604 1689 1 Standard parser parameters.
: 1605 1690 1
: 1606 1691 1 IMPLICIT INPUTS:
: 1607 1692 1 NONE
: 1608 1693 1
: 1609 1694 1 OUTPUT PARAMETERS:
: 1610 1695 1 NONE
: 1611 1696 1
: 1612 1697 1 IMPLICIT OUTPUTS:
: 1613 1698 1 NONE
: 1614 1699 1
: 1615 1700 1 ROUTINE VALUE:
: 1616 1701 1 True if /NOLOG specified.
: 1617 1702 1
: 1618 1703 1 SIDE EFFECTS:
: 1619 1704 1 NONE
: 1620 1705 1
: 1621 1706 1 !--
: 1622 1707 1
: 1623 1708 2 BEGIN
: 1624 1709 2 PARSE_EXTERNAL_REGISTERS; ! Declare external registers
: 1625 1710 2 LOCAL
: 1626 1711 2 STATUS;
: 1627 1712 2
: 1628 1713 2
: 1629 1714 2 STATUS = CLISPRESNT(.Q_DESC);
: 1630 1715 2 IF .STATUS
: 1631 1716 2 THEN
: 1632 1717 3 BEGIN
: 1633 1718 3 CLISGET_VALUE(.Q_DESC, .Q_VALUE_DESC);
: 1634 1719 3 END;
: 1635 1720 2 IF .STATUS EQL CLIS_NEGATED OR .STATUS EQL CLIS_LOCNEG
: 1636 1721 2 THEN
: 1637 1722 3 BEGIN
: 1638 1723 3 Q_ICUSOR[0,0,16,0] = 0;
: 1639 1724 3 Q_ICUSOR[2,0,16,0] = SJCS_NO_LOG_SPECIFICATION;
: 1640 1725 3 Q_ICUSOR[4,0,32,0] = 0;
: 1641 1726 3 Q_ICUSOR[8,0,32,0] = 0;
: 1642 1727 3 Q_ICUSOR = .Q_ICUSOR + 12;
: 1643 1728 3 RETURN TRUE;
: 1644 1729 2 END;
: 1645 1730 2 FALSE
: 1646 1731 1 END;
```



		04	0004 00000	.ENTRY	PARSE_LOG_FILE, Save R2	:	1680
00000000G	00		AC DD 00002	PUSHL	Q_DEST	:	1714
	52		01 FB 00005	CALLS	#T, CLISPRESENT	:	
	0C		50 D0 0000C	MOVL	R0, STATUS	:	
			52 E9 0000F	BLBC	STATUS, 1\$	:	1715
		04	58 DD 00012	PUSHL	Q_VALUE_DESC	:	1718
00000000G	00		AC DD 00014	PUSHL	Q_DESC	:	
00000000G	8F		02 FB 00017	CALLS	#2, CLISGET VALUE	:	
			52 D1 0001E 1\$:	CMPL	STATUS, #CLIS_NEGATED	:	1720
00000000G	8F		09 13 00025	BEQL	2\$	:	
			52 D1 00027	CMPL	STATUS, #CLIS_LOCNEG	:	
			0D 12 0002E	BNEQ	3\$	:	
	8B 00630000		8F D0 00030 2\$:	MOVL	#6488064, (Q_ICURSOR)+	:	1723
			8B 7C 00037	CLRQ	(Q_ICURSOR)+	:	1725
	50		01 D0 00039	MOVL	#1, R0	:	1728
			04 0003C	RET		:	
			50 D4 0003D 3\$:	CLRL	R0	:	1731
			04 0003F	RET		:	

; Routine Size: 64 bytes, Routine Base: CODE + 079A



```
: 1648      1732 1 GLOBAL ROUTINE PARSE_LOWER_UPPER(PARSE_PARAMETERS_): PARSE_LINKAGE=
: 1649      1733 1
: 1650      1734 1 !++
: 1651      1735 1
: 1652      1736 1 FUNCTIONAL DESCRIPTION:
: 1653      1737 1     This routine parses a qualifier of the form:
: 1654      1738 1         /QUAL=upper
: 1655      1739 1         /QUAL=(lower,upper)
: 1656      1740 1         /QUAL=(lower,,)
: 1657      1741 1     making an entry in the job controller parameter list.
: 1658      1742 1
: 1659      1743 1 INPUT PARAMETERS:
: 1660      1744 1     Standard parser parameters.
: 1661      1745 1
: 1662      1746 1 IMPLICIT INPUTS:
: 1663      1747 1     NONE
: 1664      1748 1
: 1665      1749 1 OUTPUT PARAMETERS:
: 1666      1750 1     NONE
: 1667      1751 1
: 1668      1752 1 IMPLICIT OUTPUTS:
: 1669      1753 1     NONE
: 1670      1754 1
: 1671      1755 1 ROUTINE VALUE:
: 1672      1756 1     NONE
: 1673      1757 1
: 1674      1758 1 SIDE EFFECTS:
: 1675      1759 1     NONE
: 1676      1760 1
: 1677      1761 1 !--
: 1678      1762 1
: 1679      1763 2 BEGIN
: 1680      1764 2 PARSE_EXTERNAL_REGISTERS;      ! Declare external registers
: 1681      1765 2 LOCAL
: 1682      1766 2     STATUS,
: 1683      1767 2     LOWER_LIMIT,
: 1684      1768 2     UPPER_LIMIT;
: 1685      1769 2
: 1686      1770 2
: 1687      1771 2 IF CLISPRESNT(.Q_DESC)
: 1688      1772 2 THEN
: 1689      1773 2     BEGIN
: 1690      1774 2     LOWER_LIMIT = 0;
: 1691      1775 2     UPPER_LIMIT = 0;
: 1692      1776 2
: 1693      1777 2
: 1694      1778 2     ! Get the first value.
: 1695      1779 2     !
: 1696      1780 2     STATUS = CLISGET VALUE(.Q_DESC, .Q_VALUE_DESC);
: 1697      1781 2     IF .Q_VALUE_DESC[DSC$W_LENGTH] NEQ 0
: 1698      1782 2     THEN
: 1699      1783 2         BEGIN
: 1700      1784 2         IF
: 1701      1785 2             BEGIN
: 1702      1786 2             IF NOT LIB$CVT DTB(
: 1703      1787 2                 .Q_VALUE_DESC[DSC$W_LENGTH], .Q_VALUE_DESC[DSC$A_POINTER],
: 1704      1788 2                 UPPER_LIMIT)
```



```
: 1705      1789  5      THEN
: 1706      1790  5      TRUE
: 1707      1791  5      ELSE
: 1708      1792  5      .UPPER_LIMIT EQL 0
: 1709      1793  5      END
: 1710      1794  4      THEN
: 1711      1795  4      SIGNAL_STOP(.Q_MESSAGE, 2, .Q_VALUE_DESC, .Q_DESC);
: 1712      1796  3      END;
: 1713      1797  3
: 1714      1798  3
: 1715      1799  3      ! Get the second value, if it exists.
: 1716      1800  3      !
: 1717      1801  3      IF .STATUS EQL CLI$_COMMA
: 1718      1802  3      THEN
: 1719      1803  4      BEGIN
: 1720      1804  4      LOWER_LIMIT = .UPPER_LIMIT;
: 1721      1805  4      UPPER_LIMIT = 0;
: 1722      1806  4      CLI$_GET_VALUE(.Q_DESC, .Q_VALUE_DESC);
: 1723      1807  4      IF .Q_VALUE_DESC[DSC$_LENGTH] NEQ 0
: 1724      1808  4      THEN
: 1725      1809  5      BEGIN
: 1726      1810  5      IF
: 1727      1811  6      BEGIN
: 1728      1812  6      IF NOT LIB$_CVT_DTB(
: 1729      1813  6      .Q_VALUE_DESC[DSC$_LENGTH], .Q_VALUE_DESC[DSC$_A_POINTER],
: 1730      1814  6      UPPER_LIMIT)
: 1731      1815  6      THEN
: 1732      1816  6      TRUE
: 1733      1817  6      ELSE
: 1734      1818  6      .UPPER_LIMIT EQL 0
: 1735      1819  6      END
: 1736      1820  5      THEN
: 1737      1821  5      SIGNAL_STOP(.Q_MESSAGE, 2, .Q_VALUE_DESC, .Q_DESC);
: 1738      1822  4      END;
: 1739      1823  3      END;
: 1740      1824  3
: 1741      1825  3
: 1742      1826  3      IF .LOWER_LIMIT NEQ 0
: 1743      1827  3      THEN
: 1744      1828  4      BEGIN
: 1745      1829  4      Q_ICURSOR[0,0,16,0] = 4;
: 1746      1830  4      Q_ICURSOR[2,0,16,0] = .Q_P1;
: 1747      1831  4      Q_ICURSOR[4,0,32,0] = .Q_DCURSOR;
: 1748      1832  4      Q_ICURSOR[8,0,32,0] = 0;
: 1749      1833  4      Q_ICURSOR = .Q_ICURSOR + 12;
: 1750      1834  4      .Q_DCURSOR = .LOWER_LIMIT;
: 1751      1835  4      Q_DCURSOR = .Q_DCURSOR + 4;
: 1752      1836  4      END
: 1753      1837  3      ELSE
: 1754      1838  4      BEGIN
: 1755      1839  4      Q_ICURSOR[0,0,16,0] = 0;
: 1756      1840  4      Q_ICURSOR[2,0,16,0] = .Q_P2;
: 1757      1841  4      Q_ICURSOR[4,0,32,0] = 0;
: 1758      1842  4      Q_ICURSOR[8,0,32,0] = 0;
: 1759      1843  4      Q_ICURSOR = .Q_ICURSOR + 12;
: 1760      1844  3      END;
: 1761      1845  3
```

```
: 1762      1846 3
: 1763      1847 3
: 1764      1848 3
: 1765      1849 4
: 1766      1850 4
: 1767      1851 4
: 1768      1852 4
: 1769      1853 4
: 1770      1854 4
: 1771      1855 4
: 1772      1856 4
: 1773      1857 4
: 1774      1858 3
: 1775      1859 4
: 1776      1860 4
: 1777      1861 4
: 1778      1862 4
: 1779      1863 4
: 1780      1864 4
: 1781      1865 3
: 1782      1866 3
: 1783      1867 3
: 1784      1868 3
: 1785      1869 3
: 1786      1870 3
: 1787      1871 3
: 1788      1872 3
: 1789      1873 2
: 1790      1874 1

      IF .UPPER_LIMIT NEQ 0
      THEN
      BEGIN
      Q_ICURSOR[0,0,16,0] = 4;
      Q_ICURSOR[2,0,16,0] = .Q_P3;
      Q_ICURSOR[4,0,32,0] = .Q_DCURSOR;
      Q_ICURSOR[8,0,32,0] = 0;
      Q_ICURSOR = .Q_ICURSOR + 12;
      .Q_DCURSOR = .UPPER_LIMIT;
      Q_DCURSOR = .Q_DCURSOR + 4;
      END
      ELSE
      BEGIN
      Q_ICURSOR[0,0,16,0] = 0;
      Q_ICURSOR[2,0,16,0] = .Q_P4;
      Q_ICURSOR[4,0,32,0] = 0;
      Q_ICURSOR[8,0,32,0] = 0;
      Q_ICURSOR = .Q_ICURSOR + 12;
      END;

      ! Try to get another value, to ensure there are not three.
      !
      IF CLISGET_VALUE(.Q_DESC, .Q_VALUE_DESC)
      THEN
      SIGNAL_STOP(.Q_MESSAGE, 2, .Q_VALUE_DESC, .Q_DESC);
      END;
      END;
```

			00FC 00000	.ENTRY	PARSE_LOWER_UPPER, Save R2,R3,R4,R5,R6,R7	: 1732
	57	00000000G	00 9E 00002	MOVAB	LIB\$CVT_DTB, R7	
	56	00000000G	00 9E 00009	MOVAB	LIB\$STOP, R6	
	55	00000000G	00 9E 00010	MOVAB	CLISGET_VALUE, R5	
	5E		04 C2 00017	SUBL2	#4, SP	
	52	04	AC D0 0001A	MOVL	Q_DESC, R2	: 1771
			52 DD 0001E	PUSHL	R2	
	00000000G	00	01 FB 00020	CALLS	#1, CLISPRESENT	
		01	50 E8 00027	BLBS	R0, 1\$	
			04 0002A	RET		
			53 D4 0002B 1\$:	CLRL	LOWER_LIMIT	: 1774
			6E D4 0002D	CLRL	UPPER_LIMIT	: 1775
		0104	8F BB 0002F	PUSHR	#4, R2, R8	: 1780
	65		02 FB 00033	CALLS	#2, CLISGET_VALUE	
	54		50 D0 00036	MOVL	R0, STATUS	
			68 B5 00039	TSTW	(Q_VALUE_DESC)	: 1781
			1D 13 0003B	BEQL	3\$	
			5E DD 0003D	PUSHL	SP	: 1786
		04	A8 DD 0003F	PUSHL	4(Q_VALUE_DESC)	: 1787
	7E		68 3C 00042	MOVZWL	(Q_VALUE_DESC), -(SP)	
	67		03 FB 00045	CALLS	#3, LIB\$CVT_DTB	
	04		50 E9 00048	BLBC	R0, 2\$	
			6E D5 0004B	TSTL	UPPER_LIMIT	: 1792



			0B 12 0004D	BNEQ 3\$	
			52 DD 0004F 2\$:	PUSHL R2	1795
			58 DD 00051	PUSHL Q_VALUE_DESC	
			02 DD 00053	PUSHL #2	
			59 DD 00055	PUSHL Q_MESSAGE	
			04 FB 00057	CALLS #4, LIB\$STOP	
00000000G	66		54 D1 0005A 3\$:	CMP L STATUS, #CLIS_COMMA	1801
	8F		2D 12 00061	BNEQ 5\$	
	53		6E D0 00063	MOVL UPPER_LIMIT, LOWER_LIMIT	1804
			6E D4 00066	CLRL UPPER_LIMIT	1805
		0104	8F BB 00068	PUSHR #*M<R2,R8>	1806
	65		02 FB 0006C	CALLS #2, CLISGET_VALUE	
			68 B5 0006F	TSTW (Q_VALUE_DESC)	1807
			1D 13 00071	BEQL 5\$	
			5E DD 00073	PUSHL SP	1812
		04	A8 DD 00075	PUSHL 4(Q_VALUE_DESC)	1813
	7E		68 3C 00078	MOVZWL (Q_VALUE_DESC), -(SP)	
	67		03 FB 0007B	CALLS #3, LIB\$VT_DTB	
	04		50 E9 0007E	BLBC R0, 4\$	
			6E D5 00081	TSTL UPPER_LIMIT	1818
			0B 12 00083	BNEQ 5\$	
			52 DD 00085 4\$:	PUSHL R2	1821
			58 DD 00087	PUSHL Q_VALUE_DESC	
			02 DD 00089	PUSHL #2	
			59 DD 0008B	PUSHL Q_MESSAGE	
			04 FB 0008D	CALLS #4, LIB\$STOP	
	66		53 D5 00090 5\$:	TSTL LOWER_LIMIT	1826
			14 13 00092	BEQL 6\$	
	6B		04 B0 00094	MOVW #4, (Q_ICURSOR)	1829
02	AB	08	AC B0 00097	MOVW Q_P1, 2(Q_ICURSOR)	1830
04	AB		5A D0 0009C	MOVL Q_DCURSOR, 4(Q_ICURSOR)	1831
		08	AB D4 000A0	CLRL 8(Q_ICURSOR)	1832
	8A		53 D0 000A3	MOVL LOWER_LIMIT, (Q_DCURSOR)+	1834
			0A 11 000A6	BRB 7\$	1826
			6B B4 000A8 6\$:	CLRW (Q_ICURSOR)	1839
02	AB	0C	AC B0 000AA	MOVW Q_P2, 2(Q_ICURSOR)	1840
		04	AB 7C 000AF	CLRW 4(Q_ICURSOR)	1841
	5B		0C C0 000B2 7\$:	ADDL2 #12, Q_ICURSOR	1833
			6E D5 000B5	TSTL UPPER_LIMIT	1847
			14 13 000B7	BEQL 8\$	
	6B		04 B0 000B9	MOVW #4, (Q_ICURSOR)	1850
02	AB	10	AC B0 000BC	MOVW Q_P3, 2(Q_ICURSOR)	1851
04	AB		5A D0 000C1	MOVL Q_DCURSOR, 4(Q_ICURSOR)	1852
		08	AB D4 000C5	CLRL 8(Q_ICURSOR)	1853
	8A		6E D0 000C8	MOVL UPPER_LIMIT, (Q_DCURSOR)+	1855
			0A 11 000CB	BRB 9\$	1847
			6B B4 000CD 8\$:	CLRW (Q_ICURSOR)	1860
02	AB	14	AC B0 000CF	MOVW Q_P4, 2(Q_ICURSOR)	1861
		04	AB 7C 000D4	CLRW 4(Q_ICURSOR)	1862
	5B		0C C0 000D7 9\$:	ADDL2 #12, Q_ICURSOR	1854
		0104	8F BB 000DA	PUSHR #*M<R2,R8>	1870
	65		02 FB 000DE	CALLS #2, CLISGET_VALUE	
	0B		50 E9 000E1	BLBC R0, 10\$	
			52 DD 000E4	PUSHL R2	1872
			58 DD 000E6	PUSHL Q_VALUE_DESC	
			02 DD 000E8	PUSHL #2	
			59 DD 000EA	PUSHL Q_MESSAGE	

JBCCMDPRS  
V04-000

Job Controller Command Parsing Utilities

G 5  
16-Sep-1984 00:09:18  
14-Sep-1984 12:08:34

VAX-11 Bliss-32 V4.0-742  
[CLIUTL.SRC]JBCCMDPRS.B32;1

Page 59  
(26)

66

04

FB 000EC  
04 000EF 10\$:

CALLS  
RET

#4, LIB\$STOP

:  
: 1874

; Routine Size: 240 bytes, Routine Base: CODE + 07DA



```
: 1792      1875 1 GLOBAL ROUTINE PARSE_NAME(PARSE_PARAMETERS_): PARSE_LINKAGE=
: 1793      1876 1
: 1794      1877 1 !++
: 1795      1878 1
: 1796      1879 1 FUNCTIONAL DESCRIPTION:
: 1797      1880 1     This routine parses the /NAME qualifier, making an entry in the job
: 1798      1881 1     controller parameter list.
: 1799      1882 1
: 1800      1883 1 INPUT PARAMETERS:
: 1801      1884 1     Standard parser parameters.
: 1802      1885 1
: 1803      1886 1 IMPLICIT INPUTS:
: 1804      1887 1     NONE
: 1805      1888 1
: 1806      1889 1 OUTPUT PARAMETERS:
: 1807      1890 1     NONE
: 1808      1891 1
: 1809      1892 1 IMPLICIT OUTPUTS:
: 1810      1893 1     NONE
: 1811      1894 1
: 1812      1895 1 ROUTINE VALUE:
: 1813      1896 1     NONE
: 1814      1897 1
: 1815      1898 1 SIDE EFFECTS:
: 1816      1899 1     NONE
: 1817      1900 1
: 1818      1901 1 !--
: 1819      1902 1
: 1820      1903 2 BEGIN
: 1821      1904 2 PARSE_EXTERNAL_REGISTERS;      ! Declare external registers
: 1822      1905 2
: 1823      1906 2
: 1824      1907 2 IF CLISGET_VALUE(.Q_DESC, .Q_VALUE_DESC)
: 1825      1908 2 THEN
: 1826      1909 3 BEGIN
: 1827      1910 3     IF .Q_VALUE_DESC[DSC$W_LENGTH] GTRU 39      ! 0 <= N <= 39
: 1828      1911 3     THEN
: 1829      1912 3         SIGNAL_STOP(.Q_MESSAGE, 2, .Q_VALUE_DESC, .Q_DESC);
: 1830      1913 3
: 1831      1914 3
: 1832      1915 3     Q_ICURSOR[0,0,16,0] = .Q_VALUE_DESC[DSC$W_LENGTH];
: 1833      1916 3     Q_ICURSOR[2,0,16,0] = SJT$ JOB_NAME;
: 1834      1917 3     Q_ICURSOR[4,0,32,0] = .Q_DCURSOR;
: 1835      1918 3     Q_ICURSOR[8,0,32,0] = 0;
: 1836      1919 3     Q_ICURSOR = .Q_ICURSOR + 12;
: 1837      1920 3     Q_DCURSOR = CH$MOVE(
: 1838      1921 3         .Q_VALUE_DESC[DSC$W_LENGTH],
: 1839      1922 3         .Q_VALUE_DESC[DSC$A_POINTER],
: 1840      1923 3         .Q_DCURSOR);
: 1841      1924 2     END;
: 1842      1925 1 END;
```

003C 00000

.ENTRY PARSE\_NAME, Save R2,R3,R4,R5

: 1875

			58	DD	00002	PUSHL	Q_VALUE_DESC	:	1907
		04	AC	DD	00004	PUSHL	Q_DESC	:	
00000000G	00		02	FB	00007	CALLS	#2, CLISGET_VALUE	:	
	29		50	E9	0000E	BLBC	R0, 2\$	:	
	27		68	B1	00011	CMPL	(Q_VALUE_DESC), #39	:	1910
			10	1B	00014	BLEQU	1\$	:	
		04	AC	DD	00016	PUSHL	Q_DESC	:	1912
			58	DD	00019	PUSHL	Q_VALUE_DESC	:	
			02	DD	0001B	PUSHL	#2	:	
			59	DD	0001D	PUSHL	Q_MESSAGE	:	
00000000G	00		04	FB	0001F	CALLS	#4, LIB\$STOP	:	
	8B		68	B0	00026	MOVW	(Q_VALUE_DESC), (Q_ICURSOR)+	:	1915
	8B	4F	8F	9B	00029	MOVZBW	#79, (Q_ICURSOR)+	:	1916
	8B		5A	D0	0002D	MOVL	Q_DCURSOR, (Q_ICURSOR)+	:	1917
			8B	D4	00030	CLRL	(Q_ICURSOR)+	:	1918
6A	04	B8	68	28	00032	MOVC3	(Q_VALUE_DESC), @4(Q_VALUE_DESC), -	:	1923
							(Q_DCURSOR)	:	
	5A		53	D0	00037	MOVL	R3, Q_DCURSOR	:	
			04	0003A	2\$:	RET		:	1925

; Routine Size: 59 bytes, Routine Base: CODE + 08CA



```
1844 1926 1 GLOBAL ROUTINE PARSE_NAME_AND_LOG_FILE(NAME_DESC,LOG_FILE_DESC,DNA_DESC,MESSAGE): PARSE_LINKAGE=
1845 1927 1
1846 1928 1 !++
1847 1929 1
1848 1930 1 FUNCTIONAL DESCRIPTION:
1849 1931 1 This routine finishes processing of the /NAME and /LOG_FILE qualifiers,
1850 1932 1 making entries in the job controller parameter list.
1851 1933 1
1852 1934 1 INPUT PARAMETERS:
1853 1935 1 NAME_DESC - Descriptor for the /NAME qualifier.
1854 1936 1 LOG_FILE_DESC - Descriptor for the /LOG_FILE qualifier.
1855 1937 1 DNA_DESC - Descriptor for the default filename string.
1856 1938 1 MESSAGE - 'Invalid log file specification' message.
1857 1939 1
1858 1940 1 IMPLICIT INPUTS:
1859 1941 1 NONE
1860 1942 1
1861 1943 1 OUTPUT PARAMETERS:
1862 1944 1 NONE
1863 1945 1
1864 1946 1 IMPLICIT OUTPUTS:
1865 1947 1 NONE
1866 1948 1
1867 1949 1 ROUTINE VALUE:
1868 1950 1 NONE
1869 1951 1
1870 1952 1 SIDE EFFECTS:
1871 1953 1 NONE
1872 1954 1
1873 1955 1 !--
1874 1956 1
1875 1957 2 BEGIN
1876 1958 2 MAP
1877 1959 2 NAME_DESC: REF BLOCK[,BYTE],
1878 1960 2 LOG_FILE_DESC: REF BLOCK[,BYTE],
1879 1961 2 DNA_DESC: REF BLOCK[,BYTE];
1880 1962 2
1881 1963 2 PARSE_EXTERNAL_REGISTERS; ! Declare external registers
1882 1964 2
1883 1965 2 LOCAL
1884 1966 2 FAB: $FAB_DECL, ! FAB for $PARSE
1885 1967 2 NAM: $NAM_DECL, ! NAM for $PARSE
1886 1968 2 ESA: VECTOR[NAM$C_MAXRSS,BYTE], ! Expanded string
1887 1969 2 STATUS;
1888 1970 2
1889 1971 2
1890 1972 2 ! Initialize RMS structures required to do a $PARSE.
1891 1973 2 !
1892 1974 2 $FAB_INIT(FAB=FAB,
1893 1975 2 DNA=.DNA_DESC[DSC$A_POINTER],
1894 1976 2 DNS=.DNA_DESC[DSC$W_LENGTH],
1895 1977 2 NAM=NAM);
1896 1978 2 $NAM_INIT(NAM=NAM,
1897 1979 2 ESA=ESA,
1898 1980 2 ESS=NAM$C_MAXRSS);
1899 1981 2 NAM[NAM$V_NOCONCEAL] = TRUE;
1900 1982 2
```

```
1901 1983 2
1902 1984 3
1903 1985 4
1904 1986 5
1905 1987 6
1906 1988 7
1907 1989 8
1908 1990 9
1909 1991 10
1910 1992 11
1911 1993 12
1912 1994 13
1913 1995 14
1914 1996 15
1915 1997 16
1916 1998 17
1917 1999 18
1918 2000 19
1919 2001 20
1920 2002 21
1921 2003 22
1922 2004 23
1923 2005 24
1924 2006 25
1925 2007 26
1926 2008 27
1927 2009 28
1928 2010 29
1929 2011 30
1930 2012 31
1931 2013 32
1932 2014 33
1933 2015 34
1934 2016 35
1935 2017 36
1936 2018 37
1937 2019 38
1938 2020 39
1939 2021 40
1940 2022 41
1941 2023 42
1942 2024 43
1943 2025 44
1944 2026 45
1945 2027 46
1946 2028 47
1947 2029 48
1948 2030 49
1949 2031 50
1950 2032 51
1951 2033 52
1952 2034 53
1953 2035 54
1954 2036 55
1955 2037 56

: Establish the primary file name. If the /LOG qualifier was used, it comes
: from that; otherwise, if the /NAME qualifier was used, it comes from that.
: If neither was used, do nothing, and let defaulting to the command file name
: happen.
IF .LOG_FILE_DESC[DSC$W_LENGTH] NEQ 0
THEN
BEGIN
FAB[FAB$B_FNS] = .LOG_FILE_DESC[DSC$W_LENGTH];
FAB[FAB$L_FNA] = .LOG_FILE_DESC[DSC$A_POINTER];
END
ELSE IF .NAME_DESC[DSC$W_LENGTH] NEQ 0
THEN
BEGIN
FAB[FAB$B_FNS] = .NAME_DESC[DSC$W_LENGTH];
FAB[FAB$L_FNA] = .NAME_DESC[DSC$A_POINTER];
END
ELSE
RETURN;

: Execute a $PARSE. Ensure that the device is a disk and that the filespec
: does not contain wildcards.
IF NOT $PARSE(FAB=FAB)
THEN
SIGNAL_STOP(.MESSAGE, 0, .FAB[FAB$L_STS], .FAB[FAB$L_STV]);

IF NOT .BBLOCK[FAB[FAB$L_DEV], DEV$V_RND]
OR .BBLOCK[FAB[FAB$L_DEV], DEV$V_FOR]
THEN
SIGNAL_STOP(.MESSAGE, 0, RMS$DEV);

IF .NAM[NAM$V_WILDCARD]
THEN
SIGNAL_STOP(.MESSAGE, 0, SHR$NOWILD OR STS$K_ERROR);

: Return the unconcealed expanded string as the log file specification.
:
Q_ICURSOR[0,0,16,0] = .NAM[NAM$B_ESL];
Q_ICURSOR[2,0,16,0] = SJCS_LOG_SPECIFICATION;
Q_ICURSOR[4,0,32,0] = .Q_DCURSOR;
Q_ICURSOR[8,0,32,0] = 0;
Q_ICURSOR = .Q_ICURSOR + 12;
Q_DCURSOR = CH$MOVE(
.NAM[NAM$B_ESL],
.NAM[NAM$L_ESA],
.Q_DCURSOR);
END;
```



				.EXTRN SYSSPARSE		
				.ENTRY		
			007C 00000		PARSE_NAME_AND_LOG_FILE, Save R2,R3,R4,R5,-	1926
					R6	
0050	8F	00	56 00000000G	00 9E 00002	MOVAB LIB\$STOP, R6	
			5E FE50	CE 9E 00009	MOVAB -432(SP), SP	
			6E	00 2C 0000E	MOVCS #0, (SP), #0, #80, \$RMS_PTR	1977
			B0	AD 00015		
			B0 AD 5003	8F B0 00017	MOVW #20483, \$RMS_PTR	
			C6 AD	02 90 0001D	MOVW #2, \$RMS_PTR+22	
			CF AD	02 90 00021	MOVW #2, \$RMS_PTR+31	
			D8 AD FF50	CD 9E 00025	MOVAB NAM, \$RMS_PTR+40	
			50 OC	AC D0 0002B	MOVL DNA_DESC, R0	
			E0 AD 04	AO D0 0002F	MOVL 4(R0), \$RMS_PTR+48	
0060	8F	00	E5 AD	60 90 00034	MOVW (R0), \$RMS_PTR+53	
			6E	00 2C 00038	MOVCS #0, (SP), #0, #96, \$RMS_PTR	1980
			FF50 CD FF50	CD 0003F		
			FF5A CD 6002	8F B0 00042	MOVW #24578, \$RMS_PTR	
			FF5C CD	01 8E 00049	MNEGB #1, \$RMS_PTR+10	
			FF58 CD	6E 9E 0004E	MOVAB ESA, \$RMS_PTR+12	
			50 08	10 88 00053	BISB2 #16, NAM+8	1981
				AC D0 00058	MOVL LOG_FILE_DESC, R0	1989
				60 B5 0005C	TSTW (R0)	
				08 12 0005E	BNEQ 1\$	
			50 04	AC D0 00060	MOVL NAME_DESC, R0	1996
				60 B5 00064	TSTW (R0)	
				66 13 00066	BEQL 6\$	
			E4 AD	60 90 00068 1\$:	MOVW (R0), FAB+52	1999
			DC AD 04	AO D0 0006C	MOVL 4(R0), FAB+44	2000
				B0 AD 9F 00071	PUSHAB FAB	2010
			00000000G	01 FB 00074	CALLS #1, SYSSPARSE	
			OC	50 E8 0007B	BLBS R0, 2\$	
			7E B8	AD 7D 0007E	MOVQ FAB+8, -(SP)	2012
				7E D4 00082	CLRL -(SP)	
			10	AC DD 00084	PUSHL MESSAGE	
			66	04 FB 00087	CALLS #4, LIB\$STOP	
04	F3		AD 04	E1 0008A 2\$:	BBC #4, FAB+67, 3\$	2015
			OE F3	AD E9 0008F	BLBC FAB+67, 4\$	2016
			000184C4	8F DD 00093 3\$:	PUSHL #99524	2018
				7E D4 00099	CLRL -(SP)	
			10	AC DD 0009B	PUSHL MESSAGE	
			66	03 FB 0009E	CALLS #3, LIB\$STOP	
			OD 85	AD E9 000A1 4\$:	BLBC NAM+53, 5\$	2021
			7E 112A	8F 3C 000A5	MOVZWL #4394, -(SP)	2023
				7E D4 000AA	CLRL -(SP)	
			10	AC DD 000AC	PUSHL MESSAGE	
			66	03 FB 000AF	CALLS #3, LIB\$STOP	
			8B FF5B	CD 9B 000B2 5\$:	MOVZBW NAM+11, (Q_ICURSOR)+	2028
			8B 62	8F 9B 000B7	MOVZBW #98, (Q_ICURSOR)+	2029
			8B	5A D0 000BB	MOVL Q_DCURSOR, (Q_ICURSOR)+	2030
				8B D4 000BE	CLRL (Q_ICURSOR)+	2031
			50 FF5B	CD 9A 000C0	MOVZBL NAM+11, R0	2034
6A	FF5C		DD 50	28 000C5	MOVCS R0, @NAM+12, (Q_DCURSOR)	2036
			5A	53 D0 000CB	MOVL R3, Q_DCURSOR	
				04 000CE 6\$:	RET	2037

JBCCMDPRS  
V04-000

Job Controller Command Parsing Utilities

M 5  
16-Sep-1984 00:09:18  
14-Sep-1984 12:08:34

VAX-11 Bliss-32 V4.0-742  
[CLIUTL.SRC]JBCCMDPRS.B32;1

Page 65  
(28)

; Routine Size: 207 bytes, Routine Base: CODE + 0905



```
1957 2038 1 GLOBAL ROUTINE PARSE_NONZERO_NUMBER(PARSE_PARAMETERS_): PARSE_LINKAGE=
1958 2039 1
1959 2040 1 ++
1960 2041 1
1961 2042 1 FUNCTIONAL DESCRIPTION:
1962 2043 1 This routine parses a qualifier with an arbitrary nonzero numeric value,
1963 2044 1 making an entry in the job controller parameter list.
1964 2045 1
1965 2046 1 INPUT PARAMETERS:
1966 2047 1 Standard parser parameters.
1967 2048 1
1968 2049 1 IMPLICIT INPUTS:
1969 2050 1 NONE
1970 2051 1
1971 2052 1 OUTPUT PARAMETERS:
1972 2053 1 NONE
1973 2054 1
1974 2055 1 IMPLICIT OUTPUTS:
1975 2056 1 NONE
1976 2057 1
1977 2058 1 ROUTINE VALUE:
1978 2059 1 NONE
1979 2060 1
1980 2061 1 SIDE EFFECTS:
1981 2062 1 NONE
1982 2063 1
1983 2064 1 --
1984 2065 1
1985 2066 2 BEGIN
1986 2067 2 PARSE_EXTERNAL_REGISTERS; ! Declare external registers
1987 2068 2
1988 2069 2
1989 2070 2 IF CLISGET_VALUE(.Q_DESC, .Q_VALUE_DESC)
1990 2071 2 THEN
1991 2072 3 BEGIN
1992 2073 3 IF
1993 2074 4 BEGIN
1994 2075 4 IF NOT LIB$CVT_DTB(
1995 2076 4 .Q_VALUE_DESC[DSC$W_LENGTH], .Q_VALUE_DESC[DSC$A_POINTER],
1996 2077 4 .Q_DCUSOR)
1997 2078 4 THEN
1998 2079 4 TRUE
1999 2080 4 ELSE
2000 2081 4 .Q_DCUSOR[0,0,32,0] EQL 0
2001 2082 4 END
2002 2083 3 THEN
2003 2084 3 SIGNAL_STOP(.Q_MESSAGE, 2, .Q_VALUE_DESC, .Q_DESC);
2004 2085 3
2005 2086 3
2006 2087 3 Q_ICUSOR[0,0,16,0] = 4;
2007 2088 3 Q_ICUSOR[2,0,16,0] = .Q_P1;
2008 2089 3 Q_ICUSOR[4,0,32,0] = .Q_DCUSOR;
2009 2090 3 Q_ICUSOR[8,0,32,0] = 0;
2010 2091 3 Q_ICUSOR = .Q_ICUSOR + 12;
2011 2092 3 Q_DCUSOR = .Q_DCUSOR + 4;
2012 2093 3 END;
2013 2094 1 END;
```



			0000 00000	.ENTRY	PARSE NONZERO_NUMBER, Save nothing	: 2038
			58 DD 00002	PUSHL	Q_VALUE_DESC	: 2070
00000000G	00	04	AC DD 00004	PUSHL	Q_DESC	
	32		02 FB 00007	CALLS	#2, CLISGET_VALUE	
			50 E9 0000E	BLBC	R0, 3\$	
			5A DD 00011	PUSHL	Q_DCURSOR	: 2077
		04	A8 DD 00013	PUSHL	4(Q_VALUE_DESC)	: 2076
	7E		68 3C 00016	MOVZWL	(Q_VALUE_DESC), -(SP)	
00000000G	00		03 FB 00019	CALLS	#3, LIB\$CVT_DTB	
	04		50 E9 00020	BLBC	R0, 1\$	
			6A D5 00023	TSTL	(Q_DCURSOR)	: 2081
			10 12 00025	BNEQ	2\$	
		04	AC DD 00027 1\$:	PUSHL	Q_DESC	: 2084
			58 DD 0002A	PUSHL	Q_VALUE_DESC	
			02 DD 0002C	PUSHL	#2	
			59 DD 0002E	PUSHL	Q_MESSAGE	
00000000G	00		04 FB 00030	CALLS	#4, LIB\$STOP	
	88		04 B0 00037 2\$:	MOVW	#4, (Q_ICURSOR)+	: 2087
	88	08	AC B0 0003A	MOVW	Q_P1, 7(Q_ICURSOR)+	: 2088
	88		8A DE 0003E	MOVAL	(Q_DCURSOR)+, (Q_ICURSOR)+	: 2089
			8B D4 00041	CLRL	(Q_ICURSOR)+	: 2090
			04 00043 3\$:	RET		: 2094

; Routine Size: 68 bytes, Routine Base: CODE + 09D4



```
: 2015      2095 1 GLOBAL ROUTINE PARSE_NUMBER(PARSE_PARAMETERS_): PARSE_LINKAGE=
: 2016      2096 1
: 2017      2097 1 !++
: 2018      2098 1
: 2019      2099 1 FUNCTIONAL DESCRIPTION:
: 2020      2100 1     This routine parses a qualifier with an arbitrary numeric value,
: 2021      2101 1     making an entry in the job controller parameter list.
: 2022      2102 1
: 2023      2103 1 INPUT PARAMETERS:
: 2024      2104 1     Standard parser parameters.
: 2025      2105 1
: 2026      2106 1 IMPLICIT INPUTS:
: 2027      2107 1     NONE
: 2028      2108 1
: 2029      2109 1 OUTPUT PARAMETERS:
: 2030      2110 1     NONE
: 2031      2111 1
: 2032      2112 1 IMPLICIT OUTPUTS:
: 2033      2113 1     NONE
: 2034      2114 1
: 2035      2115 1 ROUTINE VALUE:
: 2036      2116 1     NONE
: 2037      2117 1
: 2038      2118 1 SIDE EFFECTS:
: 2039      2119 1     NONE
: 2040      2120 1
: 2041      2121 1 !--
: 2042      2122 1
: 2043      2123 2 BEGIN
: 2044      2124 2 PARSE_EXTERNAL_REGISTERS;          ! Declare external registers
: 2045      2125 2
: 2046      2126 2
: 2047      2127 2 IF CLISGET_VALUE(.Q_DESC, .Q_VALUE_DESC)
: 2048      2128 2 THEN
: 2049      2129 2     BEGIN
: 2050      2130 2     IF NOT LIB$CVT_DTB(
: 2051      2131 2     .Q_VALUE_DESC[DSC$W_LENGTH], .Q_VALUE_DESC[DSC$A_POINTER],
: 2052      2132 2     .Q_DCUSOR)
: 2053      2133 2     THEN
: 2054      2134 2     SIGNAL_STOP(.Q_MESSAGE, 2, .Q_VALUE_DESC, .Q_DESC);
: 2055      2135 2
: 2056      2136 2
: 2057      2137 2     Q_ICUSOR[0,0,16,0] = 4;
: 2058      2138 2     Q_ICUSOR[2,0,16,0] = .Q_P1;
: 2059      2139 2     Q_ICUSOR[4,0,32,0] = .Q_DCUSOR;
: 2060      2140 2     Q_ICUSOR[8,0,32,0] = 0;
: 2061      2141 2     Q_ICUSOR = .Q_ICUSOR + 12;
: 2062      2142 2     Q_DCUSOR = .Q_DCUSOR + 4;
: 2063      2143 2     END;
: 2064      2144 1 END;
```

0000 00000  
58 DD 00002.ENTRY PARSE\_NUMBER, Save nothing  
PUSHL Q\_VALUE\_DESC: 2095  
: 2127

00000000G	00	04	AC	DD	00004	PUSHL	Q_DESC	:
	2E		02	FB	00007	CALLS	#2, CLISGET_VALUE	:
			50	E9	0000E	BLBC	R0, 2\$	:
			5A	DD	00011	PUSHL	Q_CURSOR	2132
		04	A8	DD	00013	PUSHL	4(Q_VALUE_DESC)	2131
	7E		68	3C	00016	MOVZWL	(Q_VALUE_DESC), -(SP)	:
00000000G	00		03	FB	00019	CALLS	#3, LIB\$CVT_DTB	:
	10		50	E8	00020	BLBS	R0, 1\$	:
		04	AC	DD	00023	PUSHL	Q_DESC	2134
			58	DD	00026	PUSHL	Q_VALUE_DESC	:
			02	DD	00028	PUSHL	#2	:
			59	DD	0002A	PUSHL	Q_MESSAGE	:
00000000G	00		04	FB	0002C	CALLS	#4, LIB\$STOP	:
	8B		04	B0	00033	MOVW	#4, (Q_CURSOR)+	2137
	8B	08	AC	B0	00036	MOVW	Q_P1, (Q_CURSOR)+	2138
	8B		8A	DE	0003A	MOVAL	(Q_CURSOR)+, (Q_CURSOR)+	2139
			8B	D4	0003D	CLRL	(Q_CURSOR)+	2140
			04	0003F	2\$:	RET		2144

; Routine Size: 64 bytes, Routine Base: CODE + 0A18



```
: 2066      2145 1 GLOBAL ROUTINE PARSE_OBJECT_NAME(PARSE_PARAMETERS_): PARSE_LINKAGE=
: 2067      2146 1
: 2068      2147 1 !++
: 2069      2148 1
: 2070      2149 1 FUNCTIONAL DESCRIPTION:
: 2071      2150 1     This routine parses an object name, making an entry in
: 2072      2151 1     the job controller parameter list.
: 2073      2152 1
: 2074      2153 1 INPUT PARAMETERS:
: 2075      2154 1     Standard parser parameters.
: 2076      2155 1
: 2077      2156 1 IMPLICIT INPUTS:
: 2078      2157 1     NONE
: 2079      2158 1
: 2080      2159 1 OUTPUT PARAMETERS:
: 2081      2160 1     NONE
: 2082      2161 1
: 2083      2162 1 IMPLICIT OUTPUTS:
: 2084      2163 1     NONE
: 2085      2164 1
: 2086      2165 1 ROUTINE VALUE:
: 2087      2166 1     NONE
: 2088      2167 1
: 2089      2168 1 SIDE EFFECTS:
: 2090      2169 1     NONE
: 2091      2170 1
: 2092      2171 1 --
: 2093      2172 1
: 2094      2173 2 BEGIN
: 2095      2174 2 PARSE_EXTERNAL_REGISTERS;      ! Declare external registers
: 2096      2175 2
: 2097      2176 2
: 2098      2177 2 IF CLISGET_VALUE(.Q_DESC, .Q_VALUE_DESC)
: 2099      2178 2 THEN
: 2100      2179 2     BEGIN
: 2101      2180 3     IF
: 2102      2181 4     BEGIN
: 2103      2182 4     IF NOT CALL_TPASE(.Q_VALUE_DESC, SYMB_STATES, SYMB_KEYS)
: 2104      2183 4     THEN
: 2105      2184 4     TRUE
: 2106      2185 4     ELSE
: 2107      2186 4     .Q_VALUE_DESC[DSC$W_LENGTH] - 1 GTRU 31 - 1 ! 1 <= N <= 31
: 2108      2187 4     END
: 2109      2188 3     THEN
: 2110      2189 3     SIGNAL_STOP(.Q_MESSAGE, 2, .Q_VALUE_DESC, .Q_DESC);
: 2111      2190 3
: 2112      2191 3
: 2113      2192 3     Q_ICURSOR[0,0,16,0] = .Q_VALUE_DESC[DSC$W_LENGTH];
: 2114      2193 3     Q_ICURSOR[2,0,16,0] = .Q_P1;
: 2115      2194 3     Q_ICURSOR[4,0,32,0] = .Q_DCURSOR;
: 2116      2195 3     Q_ICURSOR[8,0,32,0] = 0;
: 2117      2196 3     Q_ICURSOR = .Q_ICURSOR + 12;
: 2118      2197 3     Q_DCURSOR = CH$MOVE(
: 2119      2198 3     .Q_VALUE_DESC[DSC$W_LENGTH],
: 2120      2199 3     .Q_VALUE_DESC[DSC$A_POINTER],
: 2121      2200 3     .Q_DCURSOR);
: 2122      2201 2     END;
```

; 2123

2202 1 END;

			003C 00000	.ENTRY	PARSE_OBJECT_NAME, Save R2,R3,R4,R5	: 2145
			58 DD 00002	PUSHL	Q_VALUE_DESC	: 2177
		04	AC DD 00004	PUSHL	Q_DESC	
00000000G	00		02 FB 00007	CALLS	#2, CLISGET_VALUE	
	40		50 E9 0000E	BLBC	R0, 3\$	
		0000V	CF 9F 00011	PUSHAB	SYMB_KEYS	: 2182
		0000V	CF 9F 00015	PUSHAB	SYMB_STATES	
			58 DD 00019	PUSHL	Q_VALUE_DESC	
F594	CF		03 FB 0001B	CALLS	#3, CALL_TPARSE	
	0A		50 E9 00020	BLBC	R0, 1\$	
	50		68 3C 00023	MOVZWL	(Q_VALUE_DESC), R0	: 2186
			50 D7 00026	DECL	R0	
	1E		50 D1 00028	CMPL	R0, #30	
			10 1B 0002B	BLEQU	2\$	
		04	AC DD 0002D 1\$:	PUSHL	Q_DESC	: 2189
			58 DD 00030	PUSHL	Q_VALUE_DESC	
			02 DD 00032	PUSHL	#2	
			59 DD 00034	PUSHL	Q_MESSAGE	
00000000G	00		04 FB 00036	CALLS	#4, LIB\$STOP	
	8B		68 B0 0003D 2\$:	MOVW	(Q_VALUE_DESC), (Q_ICURSOR)+	: 2192
	8B	08	AC B0 00040	MOVW	Q_P1, (Q_ICURSOR)+	: 2193
	8B		5A D0 00044	MOVL	Q_DCURSOR, (Q_ICURSOR)+	: 2194
			8B D4 00047	CLRL	(Q_ICURSOR)+	: 2195
6A	04	B8	68 28 00049	MOV3	(Q_VALUE_DESC), @4(Q_VALUE_DESC), -	: 2200
					(Q_DCURSOR)	
	5A		53 D0 0004E	MOVL	R3, Q_DCURSOR	
			04 00051 3\$:	RET		: 2202

; Routine Size: 82 bytes, Routine Base: CODE + 0A58



```
: 2125      2203 1 GLOBAL ROUTINE PARSE_ON(PARSE_PARAMETERS_): PARSE_LINKAGE=
: 2126      2204 1
: 2127      2205 1 |++
: 2128      2206 1 |
: 2129      2207 1 | FUNCTIONAL DESCRIPTION:
: 2130      2208 1 |     This routine parses the /ON qualifier, making an entry in the
: 2131      2209 1 |     job controller parameter list.
: 2132      2210 1 |
: 2133      2211 1 | INPUT PARAMETERS:
: 2134      2212 1 |     Standard parser parameters.
: 2135      2213 1 |
: 2136      2214 1 | IMPLICIT INPUTS:
: 2137      2215 1 |     NONE
: 2138      2216 1 |
: 2139      2217 1 | OUTPUT PARAMETERS:
: 2140      2218 1 |     NONE
: 2141      2219 1 |
: 2142      2220 1 | IMPLICIT OUTPUTS:
: 2143      2221 1 |     NONE
: 2144      2222 1 |
: 2145      2223 1 | ROUTINE VALUE:
: 2146      2224 1 |     NONE
: 2147      2225 1 |
: 2148      2226 1 | SIDE EFFECTS:
: 2149      2227 1 |     NONE
: 2150      2228 1 |
: 2151      2229 1 | --
: 2152      2230 1 |
: 2153      2231 2 BEGIN
: 2154      2232 2 PARSE_EXTERNAL_REGISTERS;      ! Declare external registers
: 2155      2233 2
: 2156      2234 2
: 2157      2235 2 IF CLISGET_VALUE(.Q_DESC, .Q_VALUE_DESC)
: 2158      2236 2 THEN
: 2159      2237 3     BEGIN
: 2160      2238 3     LOCAL
: 2161      2239 3     LEN,
: 2162      2240 3     ADDR:      REF VECTOR[.BYTE],
: 2163      2241 3     P;
: 2164      2242 3     BUILTIN
: 2165      2243 3     MATCHC;
: 2166      2244 3
: 2167      2245 3
: 2168      2246 3     LEN = .Q_VALUE_DESC[DSC$WLENGTH];
: 2169      2247 3     ADDR = .Q_VALUE_DESC[DSC$A_POINTER];
: 2170      2248 3     IF MATCHC(%REF(2), UPLIT BYTE ('::'), LEN, .ADDR; ... P)
: 2171      2249 3     THEN
: 2172      2250 4         BEGIN
: 2173      2251 4         LOCAL
: 2174      2252 4         NODE_LEN;
: 2175      2253 4
: 2176      2254 4         NODE_LEN = .P - .ADDR - 2;
: 2177      2255 4         IF .NODE_LEN GTRU 6
: 2178      2256 4         THEN
: 2179      2257 4             SIGNAL_STOP(.Q_MESSAGE, 2, .Q_VALUE_DESC, .Q_DESC);
: 2180      2258 4
: 2181      2259 4
```

```
: 2182      2260  4      Q_ICursor[0,0,16,0] = 6;
: 2183      2261  4      Q_ICursor[2,0,16,0] = SJCS_SCSNODE_NAME;
: 2184      2262  4      Q_ICursor[4,0,32,0] = Q_DCursor;
: 2185      2263  4      Q_ICursor[8,0,32,0] = 0;
: 2186      2264  4      Q_ICursor = Q_ICursor + 12;
: 2187      2265  4      Q_DCursor = CH$COPY(.NODE_LEN, .ADDR, %C' ', 6, .Q_DCursor);
: 2188      2266  4
: 2189      2267  4
: 2190      2268  4      LEN = .ADDR + .LEN - .P;
: 2191      2269  4      ADDR = .P;
: 2192      2270  3      END;
: 2193      2271  3
: 2194      2272  3
: 2195      2273  3      IF .LEN NEQ 0
: 2196      2274  3      THEN
: 2197      2275  4          BEGIN
: 2198      2276  4              IF .LEN GTRU 31
: 2199      2277  4              THEN
: 2200      2278  4                  SIGNAL_STOP(.Q_MESSAGE, 2, .Q_VALUE_DESC, .Q_DESC);
: 2201      2279  4
: 2202      2280  4
: 2203      2281  4          Q_ICursor[0,0,16,0] = .LEN;
: 2204      2282  4          Q_ICursor[2,0,16,0] = SJCS_DEVICE_NAME;
: 2205      2283  4          Q_ICursor[4,0,32,0] = Q_DCursor;
: 2206      2284  4          Q_ICursor[8,0,32,0] = 0;
: 2207      2285  4          Q_ICursor = Q_ICursor + 12;
: 2208      2286  4          Q_DCursor = CH$MOVE(
: 2209      2287  4              .LEN,
: 2210      2288  4              .ADDR,
: 2211      2289  4              .Q_DCursor);
: 2212      2290  3      END;
: 2213      2291  2      END;
: 2214      2292  1      END;
```

3A 3A 00AAA P.AAD: .ASCII \::\ :

				00FC 00000	.ENTRY	PARSE ON, Save R2,R3,R4,R5,R6,R7	: 2203
		5E		04 C2 00002	SUBL2	#4, SP	: 2235
				58 DD 00005	PUSHL	Q_VALUE_DESC	
			04	AC DD 00007	PUSHL	Q_DESC	
		00000000G	00	02 FB 0000A	CALLS	#2, CLISGET_VALUE	
			7D	50 E9 00011	BLBC	R0, 4\$	
			57	68 3C 00014	MOVZWL	(Q_VALUE_DESC), LEN	: 2246
			56	04 A8 D0 00017	MOVL	4(Q_VALUE_DESC), ADDR	: 2247
66		57	DF	02 39 0001B	MATCHC	#2, P.AAD, LEN, (ADDR)	: 2248
				54 DC 00021	MOVPSL	R4	
			6E	53 D0 00023	MOVL	R3, (SP)	
		3C		02 E1 00026	BBC	#2, R4, 2\$	
		53		56 C3 0002A	SUBL3	ADDR, P, R3	: 2254
				02 C2 0002E	SUBL2	#2, NODE_LEN	: 2255
			06	53 D1 00031	CMPL	NODE_LEN, #6	
				10 1B 00034	BLEQU	1\$	
			04	AC DD 00036	PUSHL	Q_DESC	: 2257



06	20	66	53	2C	00052	MOVCS	NODE_LEN, (ADDR), #32, #6, (Q_DCUSOR)	2265
		5A	53	D0	00058	MOVL	R3, Q_DCUSOR	
	50	56	57	C1	0005B	ADDL3	LEN, ADDR, R0	2268
	57	50	56	C3	0005F	SUBL3	P, R0, LEN	
		56	57	D5	00066	MOVL	P, ADDR	2269
			27	13	00068	TSTL	LEN	2273
		1F	57	D1	0006A	BEQL	4\$	
			10	1B	0006D	CPL	LEN, #31	2276
			AC	DD	0006F	BLEQU	3\$	
			58	DD	00072	PUSHL	Q_DESC	2278
			02	DD	00074	PUSHL	Q-VALUE_DESC	
			59	DD	00076	PUSHL	#2	
			04	FB	00078	PUSHL	Q_MESSAGE	
			57	B0	0007F	CALLS	#7, LIB\$STOP	
			1B	B0	00082	MOVW	LEN, (Q_ICUSOR)+	2281
			5A	D0	00085	MOVW	#27, (Q_ICUSOR)+	2282
			8B	D4	00088	MOVL	Q_DCUSOR, (Q_ICUSOR)+	2283
			57	28	0008A	CLRL	(Q_ICUSOR)+	2284
	6A	66	53	D0	0008E	MOVC3	LEN, (ADDR), (Q_DCUSOR)	2289
		5A	04	00091	4\$:	MOVL	R3, Q_DCUSOR	
						RET		2292

: Routine Size: 146 bytes, Routine Base: CODE + 0AAC

```
: 2216      2293 1 GLOBAL ROUTINE PARSE_OWNER(PARSE_PARAMETERS_): PARSE_LINKAGE=
: 2217      2294 1
: 2218      2295 1 ++
: 2219      2296 1
: 2220      2297 1 FUNCTIONAL DESCRIPTION:
: 2221      2298 1     This routine parses the /OWNER qualifier, making an entry in
: 2222      2299 1     the job controller parameter list.
: 2223      2300 1
: 2224      2301 1 INPUT PARAMETERS:
: 2225      2302 1     Standard parser parameters.
: 2226      2303 1
: 2227      2304 1 IMPLICIT INPUTS:
: 2228      2305 1     NONE
: 2229      2306 1
: 2230      2307 1 OUTPUT PARAMETERS:
: 2231      2308 1     NONE
: 2232      2309 1
: 2233      2310 1 IMPLICIT OUTPUTS:
: 2234      2311 1     NONE
: 2235      2312 1
: 2236      2313 1 ROUTINE VALUE:
: 2237      2314 1     NONE
: 2238      2315 1
: 2239      2316 1 SIDE EFFECTS:
: 2240      2317 1     NONE
: 2241      2318 1
: 2242      2319 1 --
: 2243      2320 1
: 2244      2321 2 BEGIN
: 2245      2322 2 PARSE_EXTERNAL_REGISTERS;      ! Declare external registers
: 2246      2323 2
: 2247      2324 2
: 2248      2325 2 IF CLISGET_VALUE(.Q_DESC, .Q_VALUE_DESC)
: 2249      2326 2 THEN
: 2250      2327 3 BEGIN
: 2251      2328 3     CONVERTED_UIC = 0;
: 2252      2329 3     IF NOT CALL_TPARSE(.Q_VALUE_DESC, OWNE_STATES, OWNE_KEYS)
: 2253      2330 3     THEN
: 2254      2331 3         SIGNAL_STOP(.Q_MESSAGE, 2, .Q_VALUE_DESC, .Q_DESC);
: 2255      2332 3
: 2256      2333 3
: 2257      2334 3     Q_ICURSOR[0,0,16,0] = 4;
: 2258      2335 3     Q_ICURSOR[2,0,16,0] = SJCS_OWNER_UIC;
: 2259      2336 3     Q_ICURSOR[4,0,32,0] = .Q_DCURSOR;
: 2260      2337 3     Q_ICURSOR[8,0,32,0] = 0;
: 2261      2338 3     Q_ICURSOR = .Q_ICURSOR + 12;
: 2262      2339 3     Q_DCURSOR[0,0,32,0] = .CONVERTED_UIC;
: 2263      2340 3     Q_DCURSOR = .Q_DCURSOR + 4;
: 2264      2341 2 END;
: 2265      2342 1 END;
```

0000 00000  
58 DD 00002ENTRY PARSE\_OWNER, Save nothing  
PUSHL Q\_VALUE\_DESC: 2293  
: 2325



00000000G	00	04	AC	DD	00004	PUSHL	Q_DESC	:	
	37		02	FB	00007	CALLS	#2, CLISGET_VALUE	:	
			50	E9	0000E	BLBC	R0, 2\$	:	
		0000'	CF	D4	00011	CLRL	CONVERTED_UIC	:	2328
		0000V	CF	9F	00015	PUSHAB	OWNE_KEYS	:	2329
		0000V	CF	9F	00019	PUSHAB	OWNE_STATES	:	
			58	DD	0001D	PUSHL	Q_VALUE_DESC	:	
F4AA	CF		03	FB	0001F	CALLS	#3, CALC_TPARSE	:	
	10		50	E8	00024	BLBS	R0, 1\$	:	
		04	AC	DD	00027	PUSHL	Q_DESC	:	2331
			58	DD	0002A	PUSHL	Q_VALUE_DESC	:	
			02	DD	0002C	PUSHL	#2	:	
			59	DD	0002E	PUSHL	Q_MESSAGE	:	
00000000G	00		04	FB	00030	CALLS	#4, LIB\$STOP	:	
	8B	00700004	8F	D0	00037	MOVL	#7340036, (Q_ICURSOR)+	:	2334
	8B		5A	D0	0003E	MOVL	Q_DCURSOR, (Q_ICURSOR)+	:	2336
			8B	D4	00041	CLRL	(Q_ICURSOR)+	:	2337
	8A	0000'	CF	D0	00043	MOVL	CONVERTED_UIC, (Q_DCURSOR)+	:	2339
			04	00048	2\$:	RET		:	2342

; Routine Size: 73 bytes, Routine Base: CODE + 0B3E

```
2267 2343 1 GLOBAL ROUTINE PARSE_PARAMETERS(PARSE_PARAMETERS_): PARSE_LINKAGE=
2268 2344 1
2269 2345 1 ++
2270 2346 1
2271 2347 1 FUNCTIONAL DESCRIPTION:
2272 2348 1 This routine parses the /PARAMETERS qualifier, making an entry in the
2273 2349 1 job controller parameter list.
2274 2350 1
2275 2351 1 INPUT PARAMETERS:
2276 2352 1 Standard parser parameters.
2277 2353 1
2278 2354 1 IMPLICIT INPUTS:
2279 2355 1 NONE
2280 2356 1
2281 2357 1 OUTPUT PARAMETERS:
2282 2358 1 NONE
2283 2359 1
2284 2360 1 IMPLICIT OUTPUTS:
2285 2361 1 NONE
2286 2362 1
2287 2363 1 ROUTINE VALUE:
2288 2364 1 NONE
2289 2365 1
2290 2366 1 SIDE EFFECTS:
2291 2367 1 NONE
2292 2368 1
2293 2369 1 --
2294 2370 1
2295 2371 2 BEGIN
2296 2372 2 PARSE_EXTERNAL_REGISTERS; ! Declare external registers
2297 2373 2 LOCAL
2298 2374 2 STATUS;
2299 2375 2
2300 2376 2
2301 2377 2 STATUS = CLISPRESNT(.Q_DESC);
2302 2378 2 IF .STATUS
2303 2379 2 THEN
2304 2380 3 BEGIN
2305 2381 3 LOCAL
2306 2382 3 PARAM_COUNT;
2307 2383 3
2308 2384 3
2309 2385 3 PARAM_COUNT = 0;
2310 2386 3 WHILE CLISGET_VALUE(.Q_DESC, .Q_VALUE_DESC) DO
2311 2387 4 BEGIN
2312 2388 4 IF .PARAM_COUNT GTRU 7
2313 2389 4 OR .Q_VALUE_DESC[DSC$W_LENGTH] GTRU 255 ! 0 <= N <= 255
2314 2390 4 THEN
2315 2391 4 SIGNAL_STOP(.Q_MESSAGE, 2, .Q_VALUE_DESC, .Q_DESC);
2316 2392 4
2317 2393 4
2318 2394 4 Q_ICURSOR[0,0,16,0] = .Q_VALUE_DESC[DSC$W_LENGTH];
2319 2395 4 Q_ICURSOR[2,0,16,0] = SJCS_PARAMETER_1 + .PARAM_COUNT;
2320 2396 4 Q_ICURSOR[4,0,32,0] = .Q_DCURSOR;
2321 2397 4 Q_ICURSOR[8,0,32,0] = 0;
2322 2398 4 Q_ICURSOR = .Q_ICURSOR + 12;
2323 2399 4 Q_DCURSOR = CH$MOVE(
```



```
2324      .Q_VALUE_DESC[DC$W_LENGTH],  
2325      .Q_VALUE_DESC[DC$A_POINTER],  
2326      .Q_DCURLOR);  
2327  
2328  
2329      PARAM_COUNT = .PARAM_COUNT + 1;  
2330      END;  
2331      END;  
2332      IF .STATUS EQL CLIS_NEGATED OR .STATUS EQL CLIS_LOCNEG  
2333      THEN  
2334      BEGIN  
2335      Q_ICURLOR[0,0,16,0] = 0;  
2336      Q_ICURLOR[2,0,16,0] = SJCS_NO_PARAMETERS;  
2337      Q_ICURLOR[4,0,32,0] = 0;  
2338      Q_ICURLOR[8,0,32,0] = 0;  
2339      Q_ICURLOR = .Q_ICURLOR + 12;  
2340      END;  
2341      1 END;
```

			00FC 00000	.ENTRY	PARSE_PARAMETERS, Save R2,R3,R4,R5,R6,R7	2343
		04	AC DD 00002	PUSHL	Q_DESC	2377
00000000G	00		01 FB 00005	CALLS	#T, CLISPRESENT	
	57		50 D0 0000C	MOVL	R0, STATUS	
	47		57 E9 0000F	BLBC	STATUS, 4\$	2378
			56 D4 00012	CLRL	PARAM_COUNT	2385
			58 DD 00014	PUSHL	Q_VALUE_DESC	2386
		04	AC DD 00016	PUSHL	Q_DESC	
00000000G	00		02 FB 00019	CALLS	#2, CLISGET_VALUE	
	36		50 E9 00020	BLBC	R0, 4\$	
	07		56 D1 00023	CMPL	PARAM_COUNT, #7	2388
			07 1A 00026	BGTRU	2\$	
00FF	8F		68 B1 00028	CMPL	(Q_VALUE_DESC), #255	2389
			10 1B 0002D	BLEQU	3\$	
		04	AC DD 0002F	PUSHL	Q_DESC	2391
			58 DD 00032	PUSHL	Q_VALUE_DESC	
			02 DD 00034	PUSHL	#2	
00000000G	00		59 DD 00036	PUSHL	Q_MESSAGE	
	8B		04 FB 00038	CALLS	#4, LIB\$STOP	
8B	56		68 B0 0003F	MOVW	(Q_VALUE_DESC), (Q_ICURLOR)+	2394
	8B	0077	8F A1 00042	ADDW3	#1T9, PARAM_COUNT, -(Q_ICURLOR)+	2395
			5A D0 00048	MOVL	Q_DCURLOR, (Q_ICURLOR)+	2396
6A	04	B8	8B D4 0004B	CLRL	(Q_ICURLOR)+	2397
			68 28 0004D	MOVC3	(Q_VALUE_DESC), @4(Q_VALUE_DESC), -	2402
	5A		53 D0 00052	MOVL	R3, Q_DCURLOR	
			56 D6 00055	INCL	PARAM_COUNT	2405
			BB 11 00057	BRB	1\$	2386
00000000G	8F		57 D1 00059	CMPL	STATUS, #CLIS_NEGATED	2408
			09 13 00060	BEQL	5\$	
00000000G	8F		57 D1 00062	CMPL	STATUS, #CLIS_LOCNEG	
			09 12 00069	BNEQ	6\$	
	8B	007F0000	8F D0 0006B	MOVL	#8323072, (Q_ICURLOR)+	2411
			8B 7C 00072	CLRQ	(Q_ICURLOR)+	2413

JBCCMDPRS  
V04-000

Job Controller Command Parsing Utilities

N 6  
16-Sep-1984 00:09:18  
14-Sep-1984 12:08:34

VAX-11 Bliss-32 V4.0-742  
[CLIUTL.SRC]JBCCMDPRS.B32;1

Page 79  
(34)

04 00074 6\$: RET

; 2417

; Routine Size: 117 bytes, Routine Base: CODE + 0B87



```
: 2343 2418 1 GLOBAL ROUTINE PARSE_PRINTER(PARSE_PARAMETERS_): PARSE_LINKAGE=
: 2344 2419 1
: 2345 2420 1 ++
: 2346 2421 1
: 2347 2422 1 FUNCTIONAL DESCRIPTION:
: 2348 2423 1 This routine parses the /PRINTER qualifier, making an entry in the job
: 2349 2424 1 controller parameter list.
: 2350 2425 1
: 2351 2426 1 INPUT PARAMETERS:
: 2352 2427 1 Standard parser parameters.
: 2353 2428 1
: 2354 2429 1 IMPLICIT INPUTS:
: 2355 2430 1 NONE
: 2356 2431 1
: 2357 2432 1 OUTPUT PARAMETERS:
: 2358 2433 1 NONE
: 2359 2434 1
: 2360 2435 1 IMPLICIT OUTPUTS:
: 2361 2436 1 NONE
: 2362 2437 1
: 2363 2438 1 ROUTINE VALUE:
: 2364 2439 1 NONE
: 2365 2440 1
: 2366 2441 1 SIDE EFFECTS:
: 2367 2442 1 NONE
: 2368 2443 1
: 2369 2444 1 --
: 2370 2445 1
: 2371 2446 2 BEGIN
: 2372 2447 2 PARSE_EXTERNAL_REGISTERS; ! Declare external registers
: 2373 2448 2 LOCAL
: 2374 2449 2 STATUS;
: 2375 2450 2
: 2376 2451 2
: 2377 2452 2 STATUS = CLIS$PRESENT(.Q_DESC);
: 2378 2453 2 IF .STATUS
: 2379 2454 2 THEN
: 2380 2455 2 BEGIN
: 2381 2456 2 CLIS$GET VALUE(.Q_DESC, .Q_VALUE_DESC);
: 2382 2457 2 Q_ICURSOR[0,0,16,0] = .Q_VALUE_DESC[DSC$W_LENGTH];
: 2383 2458 2 Q_ICURSOR[2,0,16,0] = SJC$ LOG_QUEUE;
: 2384 2459 2 Q_ICURSOR[4,0,32,0] = .Q_DCURSOR;
: 2385 2460 2 Q_ICURSOR[8,0,32,0] = 0;
: 2386 2461 2 Q_ICURSOR = .Q_ICURSOR + 12;
: 2387 2462 2 Q_DCURSOR = CH$MOVE(
: 2388 2463 2 .Q_VALUE_DESC[DSC$W_LENGTH],
: 2389 2464 2 .Q_VALUE_DESC[DSC$A_POINTER],
: 2390 2465 2 .Q_DCURSOR);
: 2391 2466 2 END;
: 2392 2467 2 IF .STATUS EQL CLIS$NEGATED OR .STATUS EQL CLIS$LOCNEG
: 2393 2468 2 THEN
: 2394 2469 2 BEGIN
: 2395 2470 2 Q_ICURSOR[0,0,16,0] = 0;
: 2396 2471 2 Q_ICURSOR[2,0,16,0] = SJC$ NO_LOG_SPOOL;
: 2397 2472 2 Q_ICURSOR[4,0,32,0] = 0;
: 2398 2473 2 Q_ICURSOR[8,0,32,0] = 0;
: 2399 2474 2 Q_ICURSOR = .Q_ICURSOR + 12;
```



: 2400  
: 24012475 2 END;  
2476 1 END;

			007C 00000	.ENTRY	PARSE PRINTER, Save R2,R3,R4,R5,R6	: 2418
		04	AC DD 00002	PUSHL	Q_DESC	: 2452
00000000G	00		01 FB 00005	CALLS	#T, CLISPRESENT	
	56		50 D0 0000C	MOVL	R0, STATUS	
	20		56 E9 0000F	BLBC	STATUS, 1\$	: 2453
			58 DD 00012	PUSHL	Q_VALUE_DESC	: 2456
		04	AC DD 00014	PUSHL	Q_DESC	
00000000G	00		02 FB 00017	CALLS	#2, CLISGET VALUE	
	8B		68 B0 0001E	MOVW	(Q_VALUE_DESC), (Q_ICURSOR)+	: 2457
	8B	61	8F 9B 00021	MOVZBW	#97, (Q_ICURSOR)+	: 2458
	8B		5A D0 00025	MOVL	Q_DCURSOR, (Q_ICURSOR)+	: 2459
6A	04	B8	8B D4 00028	CLRL	(Q_ICURSOR)+	: 2460
			68 28 0002A	MOV3	(Q_VALUE_DESC), @4(Q_VALUE_DESC), -	: 2465
					(Q_DCURSOR)	
	5A		53 D0 0002F	MOVL	R3, Q_DCURSOR	
00000000G	8F		56 D1 00032 1\$:	CMPL	STATUS, #CLIS_NEGATED	: 2467
			09 13 00039	BEQL	2\$	
00000000G	8F		56 D1 0003B	CMPL	STATUS, #CLIS_LOCNEG	
			09 12 00042	BNEQ	3\$	
	8B	00650000	8F D0 00044 2\$:	MOVL	#6619136, (Q_ICURSOR)+	: 2470
			8B 7C 0004B	CLRQ	(Q_ICURSOR)+	: 2472
			04 0004D 3\$:	RET		: 2476

; Routine Size: 78 bytes, Routine Base: CODE + 0BFC



```
: 2403      2477 1 GLOBAL ROUTINE PARSE_PRIORITY(PARSE_PARAMETERS_): PARSE_LINKAGE=
: 2404      2478 1
: 2405      2479 1 !++
: 2406      2480 1
: 2407      2481 1 FUNCTIONAL DESCRIPTION:
: 2408      2482 1     This routine parses the /PRIORITY qualifier, making an entry in the job
: 2409      2483 1     controller parameter list.
: 2410      2484 1
: 2411      2485 1 INPUT PARAMETERS:
: 2412      2486 1     Standard parser parameters.
: 2413      2487 1
: 2414      2488 1 IMPLICIT INPUTS:
: 2415      2489 1     NONE
: 2416      2490 1
: 2417      2491 1 OUTPUT PARAMETERS:
: 2418      2492 1     NONE
: 2419      2493 1
: 2420      2494 1 IMPLICIT OUTPUTS:
: 2421      2495 1     NONE
: 2422      2496 1
: 2423      2497 1 ROUTINE VALUE:
: 2424      2498 1     NONE
: 2425      2499 1
: 2426      2500 1 SIDE EFFECTS:
: 2427      2501 1     NONE
: 2428      2502 1
: 2429      2503 1 --
: 2430      2504 1
: 2431      2505 2 BEGIN
: 2432      2506 2 PARSE_EXTERNAL_REGISTERS;      ! Declare external registers
: 2433      2507 2
: 2434      2508 2
: 2435      2509 2 IF CLISGET_VALUE(.Q_DESC, .Q_VALUE_DESC)
: 2436      2510 2 THEN
: 2437      2511 3     BEGIN
: 2438      2512 3     IF
: 2439      2513 4         BEGIN
: 2440      2514 4         IF NOT LIB$CVT_DTB(
: 2441      2515 4             .Q_VALUE_DESC[DSC$W_LENGTH], .Q_VALUE_DESC[DSC$A_POINTER],
: 2442      2516 4             .Q_DCUSOR)
: 2443      2517 4         THEN
: 2444      2518 4             TRUE
: 2445      2519 4         ELSE
: 2446      2520 4             .Q_DCUSOR[0,0,32,0] GTRU 255      ! 0 <= N <= 255
: 2447      2521 4         END
: 2448      2522 3     THEN
: 2449      2523 3         SIGNAL_STOP(.Q_MESSAGE, 2, .Q_VALUE_DESC, .Q_DESC);
: 2450      2524 3
: 2451      2525 3
: 2452      2526 3     Q_ICUSOR[0,0,16,0] = 4;
: 2453      2527 3     Q_ICUSOR[2,0,16,0] = SJCS_PRIORITY;
: 2454      2528 3     Q_ICUSOR[4,0,32,0] = .Q_DCUSOR;
: 2455      2529 3     Q_ICUSOR[8,0,32,0] = 0;
: 2456      2530 3     Q_ICUSOR = .Q_ICUSOR + 12;
: 2457      2531 3     Q_DCUSOR = .Q_DCUSOR + 4;
: 2458      2532 2     END;
: 2459      2533 1 END;
```

			0000 00000		.ENTRY	PARSE PRIORITY, Save nothing	:	2477
			58 DD 00002		PUSHL	Q_VALUE_DESC	:	2509
		04	AC DD 00004		PUSHL	Q_DESC	:	
00000000G	00		02 FB 00007		CALLS	#2, CLISGET_VALUE	:	
	37		50 E9 0000E		BLBC	R0, 3\$	:	
			5A DD 00011		PUSHL	Q_DCURSOR	:	2516
		04	A8 DD 00013		PUSHL	4(Q_VALUE_DESC)	:	2515
	7E		68 3C 00016		MOVZWL	(Q_VALUE_DESC), -(SP)	:	
00000000G	00		03 FB 00019		CALLS	#3, LIB\$CVT_DTB	:	
	09		50 E9 00020		BLBC	R0, 1\$	:	
000000FF	8F		6A D1 00023		CML	(Q_DCURSOR), #255	:	2520
			10 1B 0002A		BLEQU	2\$	:	
		04	AC DD 0002C 1\$:		PUSHL	Q_DESC	:	2523
			58 DD 0002F		PUSHL	Q_VALUE_DESC	:	
			02 DD 00031		PUSHL	#2	:	
			59 DD 00033		PUSHL	Q_MESSAGE	:	
00000000G	00		04 FB 00035		CALLS	#4, LIB\$STOP	:	
	8B 00820004		8F D0 0003C 2\$:		MOVL	#8519684, (Q_ICURSOR)+	:	2526
	8B		8A DE 00043		MOVAL	(Q_DCURSOR)+, (Q_ICURSOR)+	:	2528
			8B D4 00046		CLRL	(Q_ICURSOR)+	:	2529
			04 00048 3\$:		RET		:	2533

; Routine Size: 73 bytes, Routine Base: CODE + 0C4A



```
: 2461      2534 1 GLOBAL ROUTINE PARSE_PROTECTION(PARSE_PARAMETERS_): PARSE_LINKAGE=
: 2462      2535 1
: 2463      2536 1 !++
: 2464      2537 1
: 2465      2538 1 FUNCTIONAL DESCRIPTION:
: 2466      2539 1     This routine parses the /PROTECTION qualifier, making an entry in
: 2467      2540 1     the job controller parameter list.
: 2468      2541 1
: 2469      2542 1 INPUT PARAMETERS:
: 2470      2543 1     Standard parser parameters.
: 2471      2544 1
: 2472      2545 1 IMPLICIT INPUTS:
: 2473      2546 1     NONE
: 2474      2547 1
: 2475      2548 1 OUTPUT PARAMETERS:
: 2476      2549 1     NONE
: 2477      2550 1
: 2478      2551 1 IMPLICIT OUTPUTS:
: 2479      2552 1     NONE
: 2480      2553 1
: 2481      2554 1 ROUTINE VALUE:
: 2482      2555 1     NONE
: 2483      2556 1
: 2484      2557 1 SIDE EFFECTS:
: 2485      2558 1     NONE
: 2486      2559 1
: 2487      2560 1 !--
: 2488      2561 1
: 2489      2562 2 BEGIN
: 2490      2563 2 PARSE_EXTERNAL_REGISTERS;      ! Declare external registers
: 2491      2564 2
: 2492      2565 2
: 2493      2566 2 IF CLISPRESNT(.Q_DESC)
: 2494      2567 2 THEN
: 2495      2568 3     BEGIN
: 2496      2569 3     TPA 1 = 0;
: 2497      2570 3     WHILE CLISGET_VALUE(.Q_DESC, .Q_VALUE_DESC) DO
: 2498      2571 4     BEGIN
: 2499      2572 4     IF NOT CALL_TPARSE(.Q_VALUE_DESC, PROT_STATES, PROT_KEYS)
: 2500      2573 4     THEN
: 2501      2574 4     SIGNAL_STOP(.Q_MESSAGE, 2, .Q_VALUE_DESC, .Q_DESC);
: 2502      2575 3     END;
: 2503      2576 3
: 2504      2577 3
: 2505      2578 3     Q_ICURSOR[0,0,16,0] = 4;
: 2506      2579 3     Q_ICURSOR[2,0,16,0] = SJCS_PROTECTION;
: 2507      2580 3     Q_ICURSOR[4,0,32,0] = .Q_DCURSOR;
: 2508      2581 3     Q_ICURSOR[8,0,32,0] = 0;
: 2509      2582 3     Q_ICURSOR = .Q_ICURSOR + 12;
: 2510      2583 3     Q_DCURSOR[0,0,16,0] = NOT .TPA 1<0,16>;
: 2511      2584 3     Q_DCURSOR[2,0,16,0] = .TPA 1<16,16>;
: 2512      2585 3     Q_DCURSOR = .Q_DCURSOR + 4;
: 2513      2586 2     END;
: 2514      2587 1 END;
```

			0000	00000	.ENTRY	PARSE_PROTECTION, Save nothing	: 2534
00000000G	00	04	AC	DD 00002	PUSHL	Q_DESC	: 2566
	4D		01	FB 00005	CALLS	#1, CLISPRESENT	
		0000'	50	E9 0000C	BLBC	R0, 3\$	
			CF	D4 0000F	CLRL	TPA 1	: 2569
		04	58	DD 00013	PUSHL	Q_VALUE_DESC	: 2570
00000000G	00		AC	DD 00015	PUSHL	Q_DESC	
	24		02	FB 00018	CALLS	#2, CLISGET_VALUE	
		0000V	50	E9 0001F	BLBC	R0, 2\$	
		0000V	CF	9F 00022	PUSHAB	PROT_KEYS	: 2572
			CF	9F 00026	PUSHAB	PROT_STATES	
F348	CF		58	DD 0002A	PUSHL	Q_VALUE_DESC	
	DF		03	FB 0002C	CALLS	#3, CALC_TPARSE	
		04	50	E8 00031	BLBS	R0, 1\$	: 2574
			AC	DD 00034	PUSHL	Q_DESC	
			58	DD 00037	PUSHL	Q_VALUE_DESC	
			02	DD 00039	PUSHL	#2	
00000000G	00		59	DD 0003B	PUSHL	Q_MESSAGE	
			04	FB 0003D	CALLS	#4, LIB\$STOP	
			CD	11 00044	BRB	1\$	: 2570
	8B	00850004	8F	D0 00046	MOVL	#8716292, (Q_ICURSOR)+	: 2578
	8B		5A	D0 0004D	MOVL	Q_DCURSOR, (Q_ICURSOR)+	: 2580
			8B	D4 00050	CLRL	(Q_ICURSOR)+	: 2581
	8A	0000'	CF	B2 00052	MCOMW	TPA_1, (Q_DCURSOR)+	: 2583
	8A	0000'	CF	B0 00057	MOVW	TPA_1+2, (Q_DCURSOR)+	: 2584
			04	0005C	RET		: 2587

; Routine Size: 93 bytes, Routine Base: CODE + 0C93



```
2516 2588 1 GLOBAL ROUTINE PARSE_QUEUE(PARSE_PARAMETERS_): PARSE_LINKAGE=
2517 2589 1
2518 2590 1 !++
2519 2591 1
2520 2592 1 FUNCTIONAL DESCRIPTION:
2521 2593 1 This routine parses a queue name qualifier, making an entry in the job
2522 2594 1 controller parameter list.
2523 2595 1
2524 2596 1 INPUT PARAMETERS:
2525 2597 1 Standard parser parameters.
2526 2598 1
2527 2599 1 IMPLICIT INPUTS:
2528 2600 1 NONE
2529 2601 1
2530 2602 1 OUTPUT PARAMETERS:
2531 2603 1 NONE
2532 2604 1
2533 2605 1 IMPLICIT OUTPUTS:
2534 2606 1 NONE
2535 2607 1
2536 2608 1 ROUTINE VALUE:
2537 2609 1 NONE
2538 2610 1
2539 2611 1 SIDE EFFECTS:
2540 2612 1 NONE
2541 2613 1
2542 2614 1 !--
2543 2615 1
2544 2616 2 BEGIN
2545 2617 2 PARSE_EXTERNAL_REGISTERS; ! Declare external registers
2546 2618 2 LOCAL
2547 2619 2 DESC: REF BBLOCK,
2548 2620 2 STATUS;
2549 2621 2 BUILTIN
2550 2622 2 ACTUALCOUNT;
2551 2623 2
2552 2624 2
2553 2625 2 DESC = .Q_VALUE_DESC;
2554 2626 2 IF NOT CLISGET_VALUE(.Q_DESC, .Q_VALUE_DESC)
2555 2627 2 THEN
2556 2628 2 IF ACTUALCOUNT() GEQU 3
2557 2629 2 THEN
2558 2630 2 DESC = .Q_P2
2559 2631 2 ELSE
2560 2632 2 RETURN;
2561 2633 2
2562 2634 2
2563 2635 2 Q_ICURSOR[0,0,16,0] = .DESC[DSC$W_LENGTH];
2564 2636 2 Q_ICURSOR[2,0,16,0] = .Q_P1;
2565 2637 2 Q_ICURSOR[4,0,32,0] = .Q_DCURSOR;
2566 2638 2 Q_ICURSOR[8,0,32,0] = 0;
2567 2639 2 Q_ICURSOR = .Q_ICURSOR + 12;
2568 2640 2 Q_DCURSOR = CHSMOVE(
2569 2641 2 .DESC[DSC$W_LENGTH],
2570 2642 2 .DESC[DSC$A_POINTER],
2571 2643 2 .Q_DCURSOR);
2572 2644 1 END;
```

		52		003C 00000	.ENTRY	PARSE QUEUE, Save R2,R3,R4,R5	:	2588
				58 D0 00002	MOVL	Q_VALUE_DESC, DESC	:	2625
				58 DD 00005	PUSHL	Q_VALUE_DESC	:	2626
			04	AC DD 00007	PUSHL	Q_DESC	:	
	00000000G	00		02 FB 0000A	CALLS	#2, CLISGET_VALUE	:	
		09		50 E8 00011	BLBS	R0, 1\$	:	
		03		6C 91 00014	CMPB	(AP), #3	:	2628
				18 1F 00017	BLSSU	2\$	:	
		52	0C	AC D0 00019	MOVL	Q_P2, DESC	:	2630
		8B		62 B0 0001D 1\$:	MOVW	(DESC), (Q_ICURSOR)+	:	2635
		8B	08	AC B0 00020	MOVW	Q_P1, (Q_ICURSOR)+	:	2636
		8B		5A D0 00024	MOVL	Q_DCURSOR, (Q_ICURSOR)+	:	2637
				8B D4 00027	CLRL	(Q_ICURSOR)+	:	2638
6A				62 28 00029	MOVC3	(DESC), @4(DESC), (Q_DCURSOR)	:	2643
	04	B2		53 D0 0002E	MOVL	R3, Q_DCURSOR	:	
		5A		04 00031 2\$:	RET		:	2644

; Routine Size: 50 bytes, Routine Base: CODE + 0CF0



```
2574 2645 1 GLOBAL ROUTINE PARSE_SEARCH_STRING(PARSE_PARAMETERS_): PARSE_LINKAGE=
2575 2646 1
2576 2647 1 !++
2577 2648 1
2578 2649 1 FUNCTIONAL DESCRIPTION:
2579 2650 1 This routine parses the value of the /SEARCH qualifier,
2580 2651 1 making an entry in the job controller parameter list.
2581 2652 1
2582 2653 1 INPUT PARAMETERS:
2583 2654 1 Standard parser parameters.
2584 2655 1
2585 2656 1 IMPLICIT INPUTS:
2586 2657 1 NONE
2587 2658 1
2588 2659 1 OUTPUT PARAMETERS:
2589 2660 1 NONE
2590 2661 1
2591 2662 1 IMPLICIT OUTPUTS:
2592 2663 1 NONE
2593 2664 1
2594 2665 1 ROUTINE VALUE:
2595 2666 1 NONE
2596 2667 1
2597 2668 1 SIDE EFFECTS:
2598 2669 1 NONE
2599 2670 1
2600 2671 1 !--
2601 2672 1
2602 2673 2 BEGIN
2603 2674 2 PARSE_EXTERNAL_REGISTERS; ! Declare external registers
2604 2675 2
2605 2676 2
2606 2677 2 IF CLISGET_VALUE(.Q_DESC, .Q_VALUE_DESC)
2607 2678 2 THEN
2608 2679 2 BEGIN
2609 2680 2 IF .Q_VALUE_DESC[DSCSW_LENGTH] GTRU 63
2610 2681 2 THEN
2611 2682 2 SIGNAL_STOP(.Q_MESSAGE, 2, .Q_VALUE_DESC, .Q_DESC);
2612 2683 2
2613 2684 2
2614 2685 2 Q_ICURSOR[0,0,16,0] = .Q_VALUE_DESC[DSCSW_LENGTH];
2615 2686 2 Q_ICURSOR[2,0,16,0] = .Q_P1;
2616 2687 2 Q_ICURSOR[4,0,32,0] = .Q_DCURSOR;
2617 2688 2 Q_ICURSOR[8,0,32,0] = 0;
2618 2689 2 Q_ICURSOR = .Q_ICURSOR + 12;
2619 2690 2 Q_DCURSOR = CH$MOVE(
2620 2691 2 .Q_VALUE_DESC[DSCSW_LENGTH],
2621 2692 2 .Q_VALUE_DESC[DSCSA_POINTER],
2622 2693 2 .Q_DCURSOR);
2623 2694 2 END;
2624 2695 1 END;
```

			58	DD	00002	PUSHL	Q_VALUE_DESC	2677
			AC	DD	00004	PUSHL	Q_DESC	
00000000G	00	04	02	FB	00007	CALLS	#2, CLISGET_VALUE	
	29		50	E9	0000E	BLBC	R0, 2\$	
	3F		68	B1	00011	CMPL	(Q_VALUE_DESC), #63	2680
			10	1B	00014	BLEQU	1\$	
		04	AC	DD	00016	PUSHL	Q_DESC	2682
			58	DD	00019	PUSHL	Q_VALUE_DESC	
			02	DD	0001B	PUSHL	#2	
			59	DD	0001D	PUSHL	Q_MESSAGE	
00000000G	00		04	FB	0001F	CALLS	#2, LIB\$STOP	
	8B		68	B0	00026	MOVW	(Q_VALUE_DESC), (Q_ICURSOR)+	2685
	8B	08	AC	B0	00029	MOVW	Q_P1, (Q_ICURSOR)+	2686
	8B		5A	D0	0002D	MOVL	Q_DCURSOR, (Q_ICURSOR)+	2687
			8B	D4	00030	CLRL	(Q_ICURSOR)+	2688
6A	04	B8	68	28	00032	MOVCL	(Q_VALUE_DESC), @4(Q_VALUE_DESC), -	2693
							(Q_DCURSOR)	
	5A		53	D0	00037	MOVL	R3, Q_DCURSOR	
			04	0003A	2\$:	RET		2695

; Routine Size: 59 bytes, Routine Base: CODE + 0D22



```
2626 1 GLOBAL ROUTINE PARSE_STRING(PARSE_PARAMETERS_): PARSE_LINKAGE=
2627 1
2628 1 !++
2629 1
2630 1 FUNCTIONAL DESCRIPTION:
2631 1 This routine parses a qualifier whose value is an arbitrary string,
2632 1 making an entry in the job controller parameter list.
2633 1
2634 1 INPUT PARAMETERS:
2635 1 Standard parser parameters.
2636 1
2637 1 IMPLICIT INPUTS:
2638 1 NONE
2639 1
2640 1 OUTPUT PARAMETERS:
2641 1 NONE
2642 1
2643 1 IMPLICIT OUTPUTS:
2644 1 NONE
2645 1
2646 1 ROUTINE VALUE:
2647 1 NONE
2648 1
2649 1 SIDE EFFECTS:
2650 1 NONE
2651 1
2652 1 !--
2653 1
2654 2 BEGIN
2655 2 PARSE_EXTERNAL_REGISTERS; ! Declare external registers
2656 2 LOCAL
2657 2 STATUS;
2658 2 BUILTIN
2659 2 ACTUALCOUNT;
2660 2
2661 2
2662 2 STATUS = CLISPRESNT(.Q_DESC);
2663 2 IF .STATUS
2664 2 THEN
2665 2 BEGIN
2666 2 CLISGET VALUE(.Q_DESC, .Q_VALUE_DESC);
2667 2 Q_ICURSOR[0,0,16,0] = .Q_VALUE_DESC[DSCSW_LENGTH];
2668 2 Q_ICURSOR[2,0,16,0] = .Q_P1;
2669 2 Q_ICURSOR[4,0,32,0] = .Q_DCURSOR;
2670 2 Q_ICURSOR[8,0,32,0] = 0;
2671 2 Q_ICURSOR = .Q_ICURSOR + 12;
2672 2 Q_DCURSOR = CHSMOVE(
2673 2 .Q_VALUE_DESC[DSCSW_LENGTH],
2674 2 .Q_VALUE_DESC[DSCSA_POINTER],
2675 2 .Q_DCURSOR);
2676 2 END;
2677 2 IF (.STATUS EQL CLIS_NEGATED OR .STATUS EQL CLIS_LOCNEG)
2678 2 AND ACTUALCOUNT() GEQU 3
2679 2 THEN
2680 2 BEGIN
2681 2 Q_ICURSOR[0,0,16,0] = 0;
2682 2 Q_ICURSOR[2,0,16,0] = .Q_P2;
```

```
: 2683      2753 3      Q_ICursor[4,0,32,0] = 0:
: 2684      2754 3      Q_ICursor[8,0,32,0] = 0:
: 2685      2755 3      Q_ICursor = .Q_ICursor + 12:
: 2686      2756 2      END;
: 2687      2757 1      END;
```

			007C 00000	.ENTRY	PARSE_STRING, Save R2,R3,R4,R5,R6	: 2696
		04	AC DD 00002	PUSHL	Q_DEST	: 2732
00000000G	00		01 FB 00005	CALLS	#T, CLISPRESNT	
	56		50 D0 0000C	MOVL	R0, STATUS	
	20		56 E9 0000F	BLBC	STATUS, 1\$	: 2733
		04	58 DD 00012	PUSHL	Q_VALUE_DESC	: 2736
			AC DD 00014	PUSHL	Q_DESC	
00000000G	00		02 FB 00017	CALLS	#2, CLISGET VALUE	
	8B		68 B0 0001E	MOVW	(Q_VALUE_DESC), (Q_ICursor)+	: 2737
	8B	08	AC B0 00021	MOVW	Q_P1, (Q_ICursor)+	: 2738
	8B		5A D0 00025	MOVL	Q_DCURSOR, (Q_ICursor)+	: 2739
			8B D4 00028	CLRL	(Q_ICursor)+	: 2740
6A 04 B8			68 28 0002A	MOVC3	(Q_VALUE_DESC), @4(Q_VALUE_DESC), -	: 2745
					(Q_DCURSOR)	
	5A		53 D0 0002F	MOVL	R3, Q_DCURSOR	
00000000G	8F		56 D1 00032 1\$:	CMPL	STATUS, #CLIS_NEGATED	: 2747
			09 13 00039	BEQL	2\$	
00000000G	8F		56 D1 0003B	CMPL	STATUS, #CLIS_LOCNEG	
			0D 12 00042	BNEQ	3\$	
	03		6C 91 00044 2\$:	CMPB	(AP), #3	: 2748
			08 1F 00047	BLSSU	3\$	
			8B B4 00049	CLRW	(Q_ICursor)+	: 2751
	8B	0C	AC B0 0004B	MOVW	Q_P2, (Q_ICursor)+	: 2752
			8B 7C 0004F	CLRQ	(Q_ICursor)+	: 2753
			04 00051 3\$:	RET		: 2757

; Routine Size: 82 bytes, Routine Base: CODE + 0D5D



```
2689 2758 1 GLOBAL ROUTINE PARSE_STRING_255(PARSE_PARAMETERS_): PARSE_LINKAGE=
2690 2759 1
2691 2760 1 !++
2692 2761 1
2693 2762 1 FUNCTIONAL DESCRIPTION:
2694 2763 1 This routine parses a qualifier whose value is an arbitrary string
2695 2764 1 of up to 255 characters, making an entry in the job controller
2696 2765 1 parameter list.
2697 2766 1
2698 2767 1 INPUT PARAMETERS:
2699 2768 1 Standard parser parameters.
2700 2769 1
2701 2770 1 IMPLICIT INPUTS:
2702 2771 1 NONE
2703 2772 1
2704 2773 1 OUTPUT PARAMETERS:
2705 2774 1 NONE
2706 2775 1
2707 2776 1 IMPLICIT OUTPUTS:
2708 2777 1 NONE
2709 2778 1
2710 2779 1 ROUTINE VALUE:
2711 2780 1 NONE
2712 2781 1
2713 2782 1 SIDE EFFECTS:
2714 2783 1 NONE
2715 2784 1
2716 2785 1 !--
2717 2786 1
2718 2787 2 BEGIN
2719 2788 2 PARSE_EXTERNAL_REGISTERS; ! Declare external registers
2720 2789 2 LOCAL
2721 2790 2 STATUS;
2722 2791 2 BUILTIN
2723 2792 2 ACTUALCOUNT;
2724 2793 2
2725 2794 2
2726 2795 2 STATUS = CLISPRESNT(.Q_DESC);
2727 2796 2 IF .STATUS
2728 2797 2 THEN
2729 2798 2 BEGIN
2730 2799 2 CLISGET_VALUE(.Q_DESC, .Q_VALUE_DESC);
2731 2800 2
2732 2801 2
2733 2802 2 IF .Q_VALUE_DESC[DSC$W_LENGTH] GTRU 255
2734 2803 2 THEN
2735 2804 2 SIGNAL_STOP(.Q_MESSAGE, 2, .Q_VALUE_DESC, .Q_DESC);
2736 2805 2
2737 2806 2
2738 2807 2 Q_ICURSOR[0,0,16,0] = .Q_VALUE_DESC[DSC$W_LENGTH];
2739 2808 2 Q_ICURSOR[2,0,16,0] = .Q_P1;
2740 2809 2 Q_ICURSOR[4,0,32,0] = .Q_DCURSOR;
2741 2810 2 Q_ICURSOR[8,0,32,0] = 0;
2742 2811 2 Q_ICURSOR = .Q_ICURSOR + 12;
2743 2812 2 Q_DCURSOR = CH$MOVE(
2744 2813 2 .Q_VALUE_DESC[DSC$W_LENGTH],
2745 2814 2 .Q_VALUE_DESC[DSC$A_POINTER],
```



```
: 2746      2815 3      .Q_DCUSOR);
: 2747      2816 2      END;
: 2748      2817 3      IF (.STATUS EQL CLIS$ NEGATED OR .STATUS EQL CLIS$_LOCNEG)
: 2749      2818 2      AND ACTUALCOUNT() GEQU 3
: 2750      2819 2      THEN
: 2751      2820 3      BEGIN
: 2752      2821 3      Q_ICUSOR[0,0,16,0] = 0;
: 2753      2822 3      Q_ICUSOR[2,0,16,0] = .Q_P2;
: 2754      2823 3      Q_ICUSOR[4,0,32,0] = 0;
: 2755      2824 3      Q_ICUSOR[8,0,32,0] = 0;
: 2756      2825 3      Q_ICUSOR = .Q_ICUSOR + 12;
: 2757      2826 2      END;
: 2758      2827 1      END;
```

			007C 00000	.ENTRY	PARSE_STRING_255, Save R2,R3,R4,R5,R6	: 2758
		04	AC DD 00002	PUSHL	Q_DESC	: 2795
00000000G	00		01 FB 00005	CALLS	#T, CLIS\$PRESENT	
	56		50 D0 0000C	MOVL	R0, STATUS	
	37		56 E9 0000F	BLBC	STATUS, 2\$	: 2796
		04	58 DD 00012	PUSHL	Q_VALUE_DESC	: 2799
			AC DD 00014	PUSHL	Q_DESC	
00000000G	00		02 FB 00017	CALLS	#2, CLIS\$GET VALUE	
	00FF		68 B1 0001E	CMPL	(Q_VALUE_DESC), #255	: 2802
		04	10 1B 00023	BLEQU	1\$	
			AC DD 00025	PUSHL	Q_DESC	: 2804
			58 DD 00028	PUSHL	Q_VALUE_DESC	
			02 DD 0002A	PUSHL	#2	
			59 DD 0002C	PUSHL	Q_MESSAGE	
00000000G	00		04 FB 0002E	CALLS	#4, LIB\$STOP	
	8B		68 B0 00035	MOVW	(Q_VALUE_DESC), (Q_ICUSOR)+	: 2807
	8B	08	AC B0 00038	MOVW	Q_P1, (Q_ICUSOR)+	: 2808
	8B		5A D0 0003C	MOVL	Q_DCUSOR, (Q_ICUSOR)+	: 2809
			8B D4 0003F	CLRL	(Q_ICUSOR)+	: 2810
6A	04	B8	68 28 00041	MOVC3	(Q_VALUE_DESC), @4(Q_VALUE_DESC), -	: 2815
					(Q_DCUSOR)	
	5A		53 D0 00046	MOVL	R3, Q_DCUSOR	
00000000G	8F		56 D1 00049	CMPL	STATUS, #CLIS\$NEGATED	: 2817
			09 13 00050	BEQL	3\$	
00000000G	8F		56 D1 00052	CMPL	STATUS, #CLIS\$_LOCNEG	
			0D 12 00059	BNEQ	4\$	
	03		6C 91 0005B	CMPL	(AP), #3	: 2818
			08 1F 0005E	BLSSU	4\$	
			8B B4 00060	CLRW	(Q_ICUSOR)+	: 2821
	8B	0C	AC B0 00062	MOVW	Q_P2, (Q_ICUSOR)+	: 2822
			8B 7C 00066	CLRW	(Q_ICUSOR)+	: 2823
			04 00068	RET		: 2827

; Routine Size: 105 bytes, Routine Base: CODE + 0DAF



```
: 2760      2828 1 GLOBAL ROUTINE PARSE_USER(PARSE_PARAMETERS_): PARSE_LINKAGE=
: 2761      2829 1
: 2762      2830 1 !++
: 2763      2831 1
: 2764      2832 1 FUNCTIONAL DESCRIPTION:
: 2765      2833 1     This routine parses the value of the /USER qualifier, making an entry
: 2766      2834 1     in the job controller parameter list.
: 2767      2835 1
: 2768      2836 1 INPUT PARAMETERS:
: 2769      2837 1     Standard parser parameters.
: 2770      2838 1
: 2771      2839 1 IMPLICIT INPUTS:
: 2772      2840 1     NONE
: 2773      2841 1
: 2774      2842 1 OUTPUT PARAMETERS:
: 2775      2843 1     NONE
: 2776      2844 1
: 2777      2845 1 IMPLICIT OUTPUTS:
: 2778      2846 1     NONE
: 2779      2847 1
: 2780      2848 1 ROUTINE VALUE:
: 2781      2849 1     NONE
: 2782      2850 1
: 2783      2851 1 SIDE EFFECTS:
: 2784      2852 1     NONE
: 2785      2853 1
: 2786      2854 1 !--
: 2787      2855 1
: 2788      2856 2 BEGIN
: 2789      2857 2 PARSE EXTERNAL REGISTERS;      ! Declare external registers
: 2790      2858 2 EXTERNAL ROUTINE
: 2791      2859 2     LGIS$SEARCHUSER;      WEAK ADDRESSING_MODE(GENERAL);
: 2792      2860 2 LOCAL
: 2793      2861 2     UAF_DESC:      VECTOR[2],      ! Descriptor for UAF buffer
: 2794      2862 2     UAF_BUFFER:      BBLOCK[UAF$C_LENGTH], ! UAF record for user
: 2795      2863 2     STATUS;
: 2796      2864 2
: 2797      2865 2
: 2798      2866 2 IF CLIS$GET_VALUE(.Q_DESC, .Q_VALUE_DESC)
: 2799      2867 2 THEN
: 2800      2868 2     BEGIN
: 2801      2869 2     IF .Q_VALUE_DESC[DSC$W_LENGTH] - 1 GTRU 12 - 1      ! 1 <= N <= 12
: 2802      2870 2     THEN
: 2803      2871 2         SIGNAL_STOP(.Q_MESSAGE, 2, .Q_VALUE_DESC, .Q_DESC);
: 2804      2872 2
: 2805      2873 2
: 2806      2874 2     UAF_DESC[0] = %ALLOCATION(UAF_BUFFER);
: 2807      2875 2     UAF_DESC[1] = UAF_BUFFER;
: 2808      2876 2     STATUS = LGIS$SEARCHUSER(.Q_VALUE_DESC, 0, UAF_DESC);
: 2809      2877 2     IF NOT .STATUS
: 2810      2878 2     THEN
: 2811      2879 2         IF .STATUS LSS 0
: 2812      2880 2         THEN
: 2813      2881 2             SIGNAL_STOP(.Q_MESSAGE, 2, .Q_VALUE_DESC, .Q_DESC)
: 2814      2882 2         ELSE
: 2815      2883 2             SIGNAL_STOP(.Q_MESSAGE, 2, .Q_VALUE_DESC, .Q_DESC, .STATUS);
: 2816      2884 2     END
```

```
2817 2885 1
2818 2886 Q_ICursor[0,0,16,0] = 25;
2819 2887 Q_ICursor[2,0,16,0] = SJCS_USER IDENTIFICATION;
2820 2888 Q_ICursor[4,0,32,0] = Q_DCursor;
2821 2889 Q_ICursor[8,0,32,0] = 0;
2822 2890 Q_ICursor = Q_ICursor + 12;
2823 2891 Q_DCursor[0,0,32,0] = .UAF_BUFFER[UAF$UIC];
2824 2892 Q_DCursor = Q_DCursor + 4;
2825 2893 Q_DCursor = CHSMOVE(
2826 2894   UAF$S_USERNAME,
2827 2895   UAF_BUFFER[UAF$T_USERNAME],
2828 2896   Q_DCursor);
2829 2897 Q_DCursor = CHSMOVE(
2830 2898   UAF$S_ACCOUNT,
2831 2899   UAF_BUFFER[UAF$T_ACCOUNT],
2832 2900   Q_DCursor);
2833 2901 Q_DCursor[0,0,8,0] = .UAF_BUFFER[UAF$B_PRI];
2834 2902 Q_DCursor = .Q_DCursor + T;
2835 2903 END;
2836 2904 1 END;
```

				007C 00000	.WEAK LGISSEARCHUSER	
				00 9E 00002	.ENTRY PARSE_USER, Save R2,R3,R4,R5,R6	2828
	56	00000000G	00	CE 9E 00009	MOVAB LIB\$STOP, R6	
	5E	FA74	58	DD 0000E	MOVAB -1420(SP), SP	
			AC	DD 00010	PUSHL Q_VALUE_DESC	2866
		04	02	FB 00013	PUSHL Q_DESC	
	00000000G	00	50	E9 0001A	CALLS #2, CLISGET_VALUE	
		76	68	3C 0001D	BLBC R0, 4\$	
		50	50	D7 00020	MOVZWL (Q_VALUE_DESC), R0	2869
		0B	50	D1 00022	DECL R0	
			0C	1B 00025	CMPL R0, #11	
			AC	DD 00027	BLEQU 1\$	
		04	58	DD 0002A	PUSHL Q_DESC	2871
			02	DD 0002C	PUSHL Q_VALUE_DESC	
			59	DD 0002E	PUSHL #2	
			04	FB 00030	PUSHL Q_MESSAGE	
	F8	AD	8F	3C 00033 1\$:	CALLS #4, LIB\$STOP	
	FC	AD	6E	9E 00039	MOVZWL #1412, UAF_DESC	2874
			AD	9F 0003D	MOVAB UAF_BUFFER, UAF_DESC+4	2875
			7E	D4 00040	PUSHAB UAF_DESC	2876
			58	DD 00042	CLRL -(SP)	
	00000000G	00	03	FB 00044	PUSHL Q_VALUE_DESC	
		20	50	E8 0004B	CALLS #3, LGISSEARCHUSER	
			50	D5 0004E	BLBS STATUS, 3\$	2877
			0E	18 00050	TSTL STATUS	2879
			AC	DD 00052	BGEQ 2\$	
		04	58	DD 00055	PUSHL Q_DESC	2881
			02	DD 00057	PUSHL Q_VALUE_DESC	
			59	DD 00059	PUSHL #2	
			04	FB 0005B	PUSHL Q_MESSAGE	
	66		0E	11 0005E	CALLS #4, LIB\$STOP	
			50	DD 00060 2\$:	BRB 3\$	
					PUSHL STATUS	2883



		04	AC	DD	00062	PUSHL	Q_DESC	:	
			58	DD	00065	PUSHL	Q_VALUE_DESC	:	
			02	DD	00067	PUSHL	#2	:	
			59	DD	00069	PUSHL	Q_MESSAGE	:	
	66		05	FB	0006B	CALLS	#5, LIB\$STOP	:	
	8B	00960019	8F	DO	0006E	3\$:	MOVL	#9830425, (Q_ICursor)+	2886
	8B		5A	DO	00075		MOVL	Q_DCursor, (Q_ICursor)+	2888
			8B	D4	00078		CLRL	(Q_ICursor)+	2889
	8A	24	AE	DO	0007A		MOVL	UAF_BUFFER+36, (Q_DCursor)+	2891
6A	04	AE	20	28	0007E	MOVC3	#32, UAF_BUFFER+4, (Q_DCursor)	2896	
		5A	53	DO	00083	MOVL	R3, Q_DCursor	:	
6A	34	AE	20	28	00086	MOVC3	#32, UAF_BUFFER+52, (Q_DCursor)	2900	
		5A	53	DO	0008B	MOVL	R3, Q_DCursor	:	
	8A	0204	CE	90	0008E	MOVB	UAF_BUFFER+516, (Q_DCursor)+	2901	
			04	00093	4\$:	RET		2904	

; Routine Size: 148 bytes, Routine Base: CODE + 0E18

```
2838 2905 1 GLOBAL ROUTINE PARSE_WORKING_SET(PARSE_PARAMETERS_): PARSE_LINKAGE=
2839 2906 1
2840 2907 1 ++
2841 2908 1
2842 2909 1 FUNCTIONAL DESCRIPTION:
2843 2910 1 This routine parses the working set qualifiers, making an entry in the
2844 2911 1 job controller parameter list.
2845 2912 1
2846 2913 1 INPUT PARAMETERS:
2847 2914 1 Standard parser parameters.
2848 2915 1
2849 2916 1 IMPLICIT INPUTS:
2850 2917 1 NONE
2851 2918 1
2852 2919 1 OUTPUT PARAMETERS:
2853 2920 1 NONE
2854 2921 1
2855 2922 1 IMPLICIT OUTPUTS:
2856 2923 1 NONE
2857 2924 1
2858 2925 1 ROUTINE VALUE:
2859 2926 1 NONE
2860 2927 1
2861 2928 1 SIDE EFFECTS:
2862 2929 1 NONE
2863 2930 1
2864 2931 1 --
2865 2932 1
2866 2933 2 BEGIN
2867 2934 2 PARSE_EXTERNAL_REGISTERS; ! Declare external registers
2868 2935 2
2869 2936 2
2870 2937 2 IF CLISGET_VALUE(.Q_DESC, .Q_VALUE_DESC)
2871 2938 2 THEN
2872 2939 3 BEGIN
2873 2940 3 IF CALL_TPARSE(.Q_VALUE_DESC, NONE_STATES, NONE_KEYS)
2874 2941 3 THEN
2875 2942 4 BEGIN
2876 2943 4 Q_ICURSOR[0,0,16,0] = 0;
2877 2944 4 Q_ICURSOR[2,0,16,0] = .Q_P2;
2878 2945 4 Q_ICURSOR[4,0,32,0] = 0;
2879 2946 4 Q_ICURSOR[8,0,32,0] = 0;
2880 2947 4 Q_ICURSOR = .Q_ICURSOR + 12;
2881 2948 4 END
2882 2949 3 ELSE
2883 2950 4 BEGIN
2884 2951 4 IF
2885 2952 5 BEGIN
2886 2953 5 IF NOT LIB$CVT DTB(
2887 2954 5 .Q_VALUE_DESC[DSC$W_LENGTH], .Q_VALUE_DESC[DSC$A_POINTER],
2888 2955 5 .Q_DCURSOR)
2889 2956 5 THEN
2890 2957 5 TRUE
2891 2958 5 ELSE
2892 2959 5 .Q_DCURSOR[0,0,32,0] GTRU 65535 ! 0 <= N <= 65535
2893 2960 5 END
2894 2961 4 THEN
```



```
: 2895      2962  4      SIGNAL_STOP(.Q_MESSAGE, 2, .Q_VALUE_DESC, .Q_DESC);
: 2896      2963  4
: 2897      2964  4
: 2898      2965  4      IF .Q_DC_CURSOR[0,0,32,0] EQL 0
: 2899      2966  4      THEN
: 2900      2967  5          BEGIN
: 2901      2968  5              Q_I_CURSOR[0,0,16,0] = 0;
: 2902      2969  5              Q_I_CURSOR[2,0,16,0] = .Q_P2;
: 2903      2970  5              Q_I_CURSOR[4,0,32,0] = 0;
: 2904      2971  5              Q_I_CURSOR[8,0,32,0] = 0;
: 2905      2972  5              Q_I_CURSOR = .Q_I_CURSOR + 12;
: 2906      2973  5          END
: 2907      2974  4      ELSE
: 2908      2975  5          BEGIN
: 2909      2976  5              Q_I_CURSOR[0,0,16,0] = 4;
: 2910      2977  5              Q_I_CURSOR[2,0,16,0] = .Q_P1;
: 2911      2978  5              Q_I_CURSOR[4,0,32,0] = .Q_DC_CURSOR;
: 2912      2979  5              Q_I_CURSOR[8,0,32,0] = 0;
: 2913      2980  5              Q_I_CURSOR = .Q_I_CURSOR + 12;
: 2914      2981  5              Q_DC_CURSOR = .Q_DC_CURSOR + 4;
: 2915      2982  4          END;
: 2916      2983  3      END;
: 2917      2984  2      END;
: 2918      2985  1      END;
```

			0000	00000	.ENTRY	PARSE WORKING_SET, Save nothing	: 2905
			58	DD 00002	PUSHL	Q_VALUE_DESC	: 2937
		04	AC	DD 00004	PUSHL	Q_DESC	
00000000G	00		02	FB 00007	CALLS	#2, CLISGET_VALUE	
	5F		50	E9 0000E	BLBC	R0, 6\$	
		0000V	CF	9F 00011	PUSHAB	NONE_KEYS	: 2940
		0000V	CF	9F 00015	PUSHAB	NONE_STATES	
			58	DD 00019	PUSHL	Q_VALUE_DESC	
F140	CF		03	FB 0001B	CALLS	#3, CALC_TPARSE	
	2F		50	E8 00020	BLBS	R0, 3\$	
			5A	DD 00023	PUSHL	Q_DC_CURSOR	: 2955
		04	A8	DD 00025	PUSHL	4(Q_VALUE_DESC)	: 2954
	7E		68	3C 00028	MOVZWL	(Q_VALUE_DESC), -(SP)	
00000000G	00		03	FB 0002B	CALLS	#3, LIB\$CVT_DTB	
	09		50	E9 00032	BLBC	R0, 1\$	
0000FFFF	8F		6A	D1 00035	CMPL	(Q_DC_CURSOR), #65535	: 2959
			10	1B 0003C	BLEQU	2\$	
		04	AC	DD 0003E	PUSHL	Q_DESC	: 2962
			58	DD 00041	PUSHL	Q_VALUE_DESC	
			02	DD 00043	PUSHL	#2	
			59	DD 00045	PUSHL	Q_MESSAGE	
00000000G	00		04	FB 00047	CALLS	#4, LIB\$STOP	: 2965
			6A	D5 0004E	TSTL	(Q_DC_CURSOR)	
			0C	12 00050	BNEQ	4\$	
			6B	B4 00052	CLRW	(Q_I_CURSOR)	: 2968
02	AB	0C	AC	B0 00054	MOVW	Q_P2, 2(Q_I_CURSOR)	: 2969
		04	AB	7C 00059	CLRW	4(Q_I_CURSOR)	: 2970
			0F	11 0005C	BRB	5\$	: 2965

JBCCMDPRS  
V04-000

Job Controller Command Parsing Utilities

H 8  
16-Sep-1984 00:09:18  
14-Sep-1984 12:08:34

VAX-11 Bliss-32 V4.0-742  
[CLIUTL.SRC]JBCCMDPRS.B32;1

Page 99  
(43)

02	6B		04	B0	0005E	4\$:	MOVW	#4, (Q_ICURSOR)	:	2976
04	AB	08	AC	B0	00061		MOVW	Q_P1, 2(Q_ICURSOR)	:	2977
	AB		8A	DE	00066		MOVAL	(Q_DCURSOR)+, 4(Q_ICURSOR)	:	2978
		08	AB	D4	0006A		CLRL	8(Q_ICURSOR)	:	2979
	5B		0C	C0	0006D	5\$:	ADDL2	#12, Q_ICURSOR	:	2947
			04	00070	6\$:		RET		:	2985

; Routine Size: 113 bytes, Routine Base: CODE + 0EAC



```
: 2920      2986 1 $INIT STATE(NONE_STATES, NONE_KEYS);
: 2921      P 2987 1 $STATE(
: 2922      2988 1   ('NONE'));
: 2923      P 2989 1 $STATE(
: 2924      2990 1   (TPA$ EOS, TPA$ EXIT));
: 2925      2991 1
: 2926      2992 1
: 2927      2993 1 $INIT STATE(INFI_STATES, INFI_KEYS);
: 2928      P 2994 1 $STATE(
: 2929      2995 1   ('INFINITE'));
: 2930      P 2996 1 $STATE(
: 2931      2997 1   (TPA$ EOS, TPA$ EXIT));
: 2932      2998 1
: 2933      2999 1
: 2934      3000 1 $INIT STATE(SYMB_STATES, SYMB_KEYS);
: 2935      P 3001 1 $STATE(
: 2936      3002 1   (TPA$ SYMBOL));
: 2937      P 3003 1 $STATE(
: 2938      3004 1   (TPA$ EOS, TPA$ EXIT));
: 2939      3005 1
: 2940      3006 1
: 2941      3007 1 $INIT STATE(MASK_STATES, MASK_KEYS);
: 2942      P 3008 1 $STATE(
: 2943      3009 1   ('MASK'));
: 2944      P 3010 1 $STATE(
: 2945      3011 1   (TPA$ EOS, TPA$ EXIT));
: 2946      3012 1
: 2947      3013 1
: 2948      3014 1 $INIT STATE(OWNE_STATES, OWNE_KEYS);
: 2949      P 3015 1 $STATE(
: 2950      3016 1   (TPA$ IDENT, ..., CONVERTED_UIC));
: 2951      P 3017 1 $STATE(
: 2952      3018 1   (TPA$ EOS, TPA$ EXIT));
: 2953      3019 1
: 2954      3020 1
: 2955      3021 1 $INIT STATE(PROT_STATES, PROT_KEYS);
: 2956      P 3022 1 $STATE(NEXTPRO,
: 2957      P 3023 1   ('SYSTEM', SYPR, .., XX'000F0000', TPA_1),
: 2958      P 3024 1   ('OWNER', OWPR, .., XX'00F00000', TPA_1),
: 2959      P 3025 1   ('GROUP', GRPR, .., XX'0F000000', TPA_1),
: 2960      3026 1   ('WORLD', WOPR, .., XX'F0000000', TPA_1));
: 2961      P 3027 1 $STATE(SYPR,
: 2962      P 3028 1   (':'),
: 2963      P 3029 1   ('='),
: 2964      3030 1   (TPA$ LAMBDA, ENDPRO));
: 2965      P 3031 1 $STATE(SYPRO,
: 2966      P 3032 1   ('R', SYPRO, .., XX'0001', TPA_1),
: 2967      P 3033 1   ('W', SYPRO, .., XX'0002', TPA_1),
: 2968      P 3034 1   ('E', SYPRO, .., XX'0004', TPA_1),
: 2969      P 3035 1   ('D', SYPRO, .., XX'0008', TPA_1),
: 2970      3036 1   (TPA$ LAMBDA, ENDPRO));
: 2971      P 3037 1 $STATE(OWPR,
: 2972      P 3038 1   (':'),
: 2973      P 3039 1   ('='),
: 2974      3040 1   (TPA$ LAMBDA, ENDPRO));
: 2975      P 3041 1 $STATE(OWPRO,
: 2976      P 3042 1   ('R', OWPRO, .., XX'0010', TPA_1),
```

```
: 2977 P 3043 1 ('W',OWPRO,,ZX'0020',TPA_1),
: 2978 P 3044 1 ('E',OWPRO,,ZX'0040',TPA_1),
: 2979 P 3045 1 ('D',OWPRO,,ZX'0080',TPA_1),
: 2980 3046 1 (TPAS_LAMBDA,ENDPRO));
: 2981 P 3047 1 $STATE(GRPR,
: 2982 P 3048 1 (':'),
: 2983 P 3049 1 ('='),
: 2984 3050 1 (TPAS_LAMBDA,ENDPRO));
: 2985 P 3051 1 $STATE(GRPRO,
: 2986 P 3052 1 ('R',GRPRO,,ZX'0100',TPA_1),
: 2987 P 3053 1 ('W',GRPRO,,ZX'0200',TPA_1),
: 2988 P 3054 1 ('E',GRPRO,,ZX'0400',TPA_1),
: 2989 P 3055 1 ('D',GRPRO,,ZX'0800',TPA_1),
: 2990 3056 1 (TPAS_LAMBDA,ENDPRO));
: 2991 P 3057 1 $STATE(WOPR,
: 2992 P 3058 1 (':'),
: 2993 P 3059 1 ('='),
: 2994 3060 1 (TPAS_LAMBDA,ENDPRO));
: 2995 P 3061 1 $STATE(WOPRO,
: 2996 P 3062 1 ('R',WOPRO,,ZX'1000',TPA_1),
: 2997 P 3063 1 ('W',WOPRO,,ZX'2000',TPA_1),
: 2998 P 3064 1 ('E',WOPRO,,ZX'4000',TPA_1),
: 2999 P 3065 1 ('D',WOPRO,,ZX'8000',TPA_1),
: 3000 3066 1 (TPAS_LAMBDA,ENDPRO));
: 3001 P 3067 1 $STATE(ENDPRO,
: 3002 P 3068 1 ('',NEXTPRO)
: 3003 3069 1 (TPAS_EOS,TPAS_EXIT));
```



: 3005  
: 30063070 1 END  
3071 0 ELUDOM

```
.PSECT _LIB$KEY1$,NOWRT, SHR, PIC,1

00000 :TPASKEYSTO
      U.2: .BLKB 0
45 4E 4F 4E 00000 :TPASKEYST
      U.4: .ASCII \NONE\
      FF 00004 :TPASKEYFILL
      FF 00005 :TPASKEYFILL
      U.6: .BYTE -1
00006 :TPASKEYSTO
      U.10: .BLKB 0
45 54 49 4E 49 46 4E 49 00006 :TPASKEYST
      U.12: .ASCII \INFINITE\
      FF 0000E :TPASKEYFILL
      FF 0000F :TPASKEYFILL
      U.14: .BYTE -1
00010 :TPASKEYSTO
      U.22: .BLKB 0
4B 53 41 4D 00010 :TPASKEYST
      U.24: .ASCII \MASK\
      FF 00014 :TPASKEYFILL
      FF 00015 :TPASKEYFILL
      U.26: .BYTE -1
00016 :TPASKEYSTO
      U.35: .BLKB 0
4D 45 54 53 59 53 00016 :TPASKEYST
      U.37: .ASCII \SYSTEM\
      FF 0001C :TPASKEYFILL
      0001D :TPASKEYSTO
      U.43: .BLKB 0
52 45 4E 57 4F 0001D :TPASKEYST
      U.45: .ASCII \OWNER\
      FF 00022 :TPASKEYFILL
      00023 :TPASKEYSTO
      U.51: .BLKB 0
50 55 4F 52 47 00023 :TPASKEYST
      U.53: .ASCII \GROUP\
      FF 00028 :TPASKEYFILL
      00029 :TPASKEYSTO
      U.59: .BLKB 0
44 4C 52 4F 57 00029 :TPASKEYST
      U.61: .ASCII \WORLD\
      FF 0002E :TPASKEYFILL
      FF 0002F :TPASKEYFILL
      U.67: .BYTE -1

.PSECT _LIB$STATES$,NOWRT, SHR, PIC,1

00000 NONE_STATES::
      .BLKB 0
0500 00000 :TPASTYPE
      U.5: .WORD 1280
```

15F7	00002	:TPASTYPE			
		U.7:	.WORD	5623	:
FFFF	00004	:TPASTARGET			:
		U.8:	.WORD	-1	:
	00006		.BLKB	2	:
	00008	INFI_STATES::			:
			.BLKB	0	:
0500	00008	:TPASTYPE			:
		U.13:	.WORD	1280	:
15F7	0000A	:TPASTYPE			:
		U.15:	.WORD	5623	:
FFFF	0000C	:TPASTARGET			:
		U.16:	.WORD	-1	:
	0000E		.BLKB	2	:
	00010	SYMB_STATES::			:
			.BLKB	0	:
05F1	00010	:TPASTYPE			:
		U.18:	.WORD	1521	:
15F7	00012	:TPASTYPE			:
		U.19:	.WORD	5623	:
FFFF	00014	:TPASTARGET			:
		U.20:	.WORD	-1	:
	00016		.BLKB	2	:
	00018	MASK_STATES::			:
			.BLKB	0	:
0500	00018	:TPASTYPE			:
		U.25:	.WORD	1280	:
15F7	0001A	:TPASTYPE			:
		U.27:	.WORD	5623	:
FFFF	0001C	:TPASTARGET			:
		U.28:	.WORD	-1	:
	0001E		.BLKB	2	:
	00020	OWNE_STATES::			:
			.BLKB	0	:
45EC	00020	:TPASTYPE			:
		U.30:	.WORD	17900	:
00000000*	00022	:TPASADDR			:
		U.31:	.LONG	<<CONVERTED_UIC-U.31>-4>	:
15F7	00026	:TPASTYPE			:
		U.32:	.WORD	5623	:
FFFF	00028	:TPASTARGET			:
		U.33:	.WORD	-1	:
	0002A		.BLKB	2	:
	0002C	PROT_STATES::			:
			.BLKB	0	:
	0002C	NEXTPRO::	.BLKB	0	:
7100	0002C	:TPASTYPE			:
		U.38:	.WORD	28928	:
00000000*	0002E	:TPASADDR			:
		U.39:	.LONG	<<TPA_1-U.39>-4>	:
000F0000	00032	:TPASMASK			:
		U.40:	.LONG	983040	:
0000*	00036	:TPASTARGET			:
		U.42:	.WORD	<<U.41-U.42>-2>	:
7101	00038	:TPASTYPE			:
		U.46:	.WORD	28929	:
00000000*	0003A	:TPASADDR			:



00F00000	0003E	U.47: .LONG	<<TPA_1-U.47>-4>	:
		:TPASMASK		:
0000*	00042	U.48: .LONG	15728640	:
		:TPASTARGET		:
7102	00044	U.50: .WORD	<<U.49-U.50>-2>	:
		:TPASTYPE		:
00000000*	00046	U.54: .WORD	28930	:
		:TPASADDR		:
0F000000	0004A	U.55: .LONG	<<TPA_1-U.55>-4>	:
		:TPASMASK		:
0000*	0004E	U.56: .LONG	251658240	:
		:TPASTARGET		:
7503	00050	U.58: .WORD	<<U.57-U.58>-2>	:
		:TPASTYPE		:
00000000*	00052	U.62: .WORD	29955	:
		:TPASADDR		:
F0000000	00056	U.63: .LONG	<<TPA_1-U.63>-4>	:
		:TPASMASK		:
0000*	0005A	U.64: .LONG	-268435456	:
		:TPASTARGET		:
	0005C	U.66: .WORD	<<U.65-U.66>-2>	:
		:SYPR		:
003A	0005C	U.41: .BLKB	0	:
		:TPASTYPE		:
003D	0005E	U.68: .WORD	58	:
		:TPASTYPE		:
15F6	00060	U.69: .WORD	61	:
		:TPASTYPE		:
0000*	00062	U.70: .WORD	5622	:
		:TPASTARGET		:
	00064	U.72: .WORD	<<U.71-U.72>-2>	:
7052	00064	SYPRO: .BLKB	0	:
		:TPASTYPE		:
		U.73: .WORD	28754	:
00000000*	00066	:TPASADDR		:
		U.74: .LONG	<<TPA_1-U.74>-4>	:
00000001	0006A	:TPASMASK		:
		U.75: .LONG	1	:
0000*	0006E	:TPASTARGET		:
		U.76: .WORD	<<SYPRO-U.76>-2>	:
7057	00070	:TPASTYPE		:
		U.77: .WORD	28759	:
00000000*	00072	:TPASADDR		:
		U.78: .LONG	<<TPA_1-U.78>-4>	:
00000002	00076	:TPASMASK		:
		U.79: .LONG	2	:
0000*	0007A	:TPASTARGET		:
		U.80: .WORD	<<SYPRO-U.80>-2>	:
7045	0007C	:TPASTYPE		:
		U.81: .WORD	28741	:
00000000*	0007E	:TPASADDR		:
		U.82: .LONG	<<TPA_1-U.82>-4>	:
00000004	00082	:TPASMASK		:
		U.83: .LONG	4	:
0000*	00086	:TPASTARGET		:
		U.84: .WORD	<<SYPRO-U.84>-2>	:
7044	00088	:TPASTYPE		:

00000000*	0008A	U.85: .WORD	28740	:
		:TPASADDR		:
00000008	0008E	U.86: .LONG	<<TPA_1-U.86>-4>	:
		:TPASMASK		:
0000*	00092	U.87: .LONG	8	:
		:TPASTARGET		:
15F6	00094	U.88: .WORD	<<SYPRO-U.88>-2>	:
		:TPASTYPE		:
0000*	00096	U.89: .WORD	5622	:
		:TPASTARGET		:
	00098	U.90: .WORD	<<U.71-U.90>-2>	:
		:OWPR		:
003A	00098	U.49: .BLKB	0	:
		:TPASTYPE		:
003D	0009A	U.91: .WORD	58	:
		:TPASTYPE		:
15F6	0009C	U.92: .WORD	61	:
		:TPASTYPE		:
0000*	0009E	U.93: .WORD	5622	:
		:TPASTARGET		:
		U.94: .WORD	<<U.71-U.94>-2>	:
	000A0	OWPRO: .BLKB	0	:
7052	000A0	:TPASTYPE		:
		U.95: .WORD	28754	:
00000000*	000A2	:TPASADDR		:
		U.96: .LONG	<<TPA_1-U.96>-4>	:
00000010	000A6	:TPASMASK		:
		U.97: .LONG	16	:
0000*	000AA	:TPASTARGET		:
		U.98: .WORD	<<OWPRO-U.98>-2>	:
7057	000AC	:TPASTYPE		:
		U.99: .WORD	28759	:
00000000*	000AE	:TPASADDR		:
		U.100: .LONG	<<TPA_1-U.100>-4>	:
00000020	000B2	:TPASMASK		:
		U.101: .LONG	32	:
0000*	000B6	:TPASTARGET		:
		U.102: .WORD	<<OWPRO-U.102>-2>	:
7045	000B8	:TPASTYPE		:
		U.103: .WORD	28741	:
00000000*	000BA	:TPASADDR		:
		U.104: .LONG	<<TPA_1-U.104>-4>	:
00000040	000BE	:TPASMASK		:
		U.105: .LONG	64	:
0000*	000C2	:TPASTARGET		:
		U.106: .WORD	<<OWPRO-U.106>-2>	:
7044	000C4	:TPASTYPE		:
		U.107: .WORD	28740	:
00000000*	000C6	:TPASADDR		:
		U.108: .LONG	<<TPA_1-U.108>-4>	:
00000080	000CA	:TPASMASK		:
		U.109: .LONG	128	:
0000*	000CE	:TPASTARGET		:
		U.110: .WORD	<<OWPRO-U.110>-2>	:
15F6	000D0	:TPASTYPE		:
		U.111: .WORD	5622	:
0000*	000D2	:TPASTARGET		:



		U.112: .WORD	<<U.71-U.112>-2>	:
	000D4	:GRPR		
		U.57: .BLKB	0	
003A	000D4	:TPATYPE		
		U.113: .WORD	58	:
003D	000D6	:TPATYPE		:
		U.114: .WORD	61	:
15F6	000D8	:TPATYPE		:
		U.115: .WORD	5622	:
0000*	000DA	:TPATARGET		:
		U.116: .WORD	<<U.71-U.116>-2>	:
	000DC	:GRPRO: .BLKB	0	:
7052	000DC	:TPATYPE		:
		U.117: .WORD	28754	:
00000000*	000DE	:TPASADDR		:
		U.118: .LONG	<<TPA_1-U.118>-4>	:
00000100	000E2	:TPASMASK		:
		U.119: .LONG	256	:
0000*	000E6	:TPATARGET		:
		U.120: .WORD	<<GRPRO-U.120>-2>	:
7057	000E8	:TPATYPE		:
		U.121: .WORD	28759	:
00000000*	000EA	:TPASADDR		:
		U.122: .LONG	<<TPA_1-U.122>-4>	:
00000200	000EE	:TPASMASK		:
		U.123: .LONG	512	:
0000*	000F2	:TPATARGET		:
		U.124: .WORD	<<GRPRO-U.124>-2>	:
7045	000F4	:TPATYPE		:
		U.125: .WORD	28741	:
00000000*	000F6	:TPASADDR		:
		U.126: .LONG	<<TPA_1-U.126>-4>	:
00000400	000FA	:TPASMASK		:
		U.127: .LONG	1024	:
0000*	000FE	:TPATARGET		:
		U.128: .WORD	<<GRPRO-U.128>-2>	:
7044	00100	:TPATYPE		:
		U.129: .WORD	28740	:
00000000*	00102	:TPASADDR		:
		U.130: .LONG	<<TPA_1-U.130>-4>	:
00000800	00106	:TPASMASK		:
		U.131: .LONG	2048	:
0000*	0010A	:TPATARGET		:
		U.132: .WORD	<<GRPRO-U.132>-2>	:
15F6	0010C	:TPATYPE		:
		U.133: .WORD	5622	:
0000*	0010E	:TPATARGET		:
		U.134: .WORD	<<U.71-U.134>-2>	:
	00110	:WOPR		:
		U.65: .BLKB	0	:
003A	00110	:TPATYPE		:
		U.135: .WORD	58	:
003D	00112	:TPATYPE		:
		U.136: .WORD	61	:
15F6	00114	:TPATYPE		:
		U.137: .WORD	5622	:
0000*	00116	:TPATARGET		:



```

      7052 00118 U.138: .WORD <<U.71-U.138>-2> ;
      00118 WOPRO: .BLKB 0 ;
      00118 :TPASTYPE ;
00000000* 0011A U.139: .WORD 28754 ;
      :TPASADDR ;
00001000 0011E U.140: .LONG <<TPA_1-U.140>-4> ;
      :TPASMASK ;
0000* 00122 U.141: .LONG 4096 ;
      :TPASTARGET ;
      7057 00124 U.142: .WORD <<WOPRO-U.142>-2> ;
      :TPASTYPE ;
00000000* 00126 U.143: .WORD 28759 ;
      :TPASADDR ;
00002000 0012A U.144: .LONG <<TPA_1-U.144>-4> ;
      :TPASMASK ;
0000* 0012E U.145: .LONG 8192 ;
      :TPASTARGET ;
      7045 00130 U.146: .WORD <<WOPRO-U.146>-2> ;
      :TPASTYPE ;
00000000* 00132 U.147: .WORD 28741 ;
      :TPASADDR ;
00004000 00136 U.148: .LONG <<TPA_1-U.148>-4> ;
      :TPASMASK ;
0000* 0013A U.149: .LONG 16384 ;
      :TPASTARGET ;
      7044 0013C U.150: .WORD <<WOPRO-U.150>-2> ;
      :TPASTYPE ;
00000000* 0013E U.151: .WORD 28740 ;
      :TPASADDR ;
00008000 00142 U.152: .LONG <<TPA_1-U.152>-4> ;
      :TPASMASK ;
0000* 00146 U.153: .LONG 32768 ;
      :TPASTARGET ;
      15F6 00148 U.154: .WORD <<WOPRO-U.154>-2> ;
      :TPASTYPE ;
0000* 0014A U.155: .WORD 5622 ;
      :TPASTARGET ;
      0014C U.156: .WORD <<U.71-U.156>-2> ;
      :ENDPRO ;
      102C 0014C U.71: .BLKB 0 ;
      :TPASTYPE ;
0000* 0014E U.157: .WORD 4140 ;
      :TPASTARGET ;
      15F7 00150 U.158: .WORD <<NEXTPRO-U.158>-2> ;
      :TPASTYPE ;
      FFFF 00152 U.159: .WORD 5623 ;
      :TPASTARGET ;
      U.160: .WORD -1 ;
      :
      .PSECT _LIB$KEY0$,NOWRT, SHR, PIC,1
00000 NONE_KEYS::
      .BLKB 0
00000 :TPASKEY0
0000* 00000 U.1: .BLKB 0
      :TPASKEY
      U.3: .WORD <U.2-U.1> ;
```



```
00002      .BLKB      2
00004 INFI_KEYS::
00004      .BLKB      0
00004 ;TPASKEY0
00004 U.9:      .BLKB      0
0000* 00004 ;TPASKEY
00004 U.11:      .WORD      <U.10-U.9>
00006      .BLKB      2
00008 SYMB_KEYS::
00008      .BLKB      0
00008 ;TPASKEY0
00008 U.17:      .BLKB      0
00008 MASK_KEYS::
00008      .BLKB      0
00008 ;TPASKEY0
0000* 00008 U.21:      .BLKB      0
0000* 00008 ;TPASKEY
00008 U.23:      .WORD      <U.22-U.21>
0000A      .BLKB      2
0000C OWNE_KEYS::
0000C      .BLKB      0
0000C ;TPASKEY0
0000C U.29:      .BLKB      0
0000C PROT_KEYS::
0000C      .BLKB      0
0000C ;TPASKEY0
0000* 0000C U.34:      .BLKB      0
0000* 0000C ;TPASKEY
0000* 0000E U.36:      .WORD      <U.35-U.34>
0000* 0000E ;TPASKEY
0000* 00010 U.44:      .WORD      <U.43-U.34>
0000* 00010 ;TPASKEY
0000* 00012 U.52:      .WORD      <U.51-U.34>
0000* 00012 ;TPASKEY
0000*      U.60:      .WORD      <U.59-U.34>
```

.EXTRN LIB\$STOP

## PSECT SUMMARY

Name	Bytes	Attributes
CODE	3869	NOVEC,NOWRT, RD, EXE,NOSHR, LCL, REL, CON,NOPI,ALIGN(2)
DATA	8	NOVEC, WRT, RD, NOEXE,NOSHR, LCL, REL, CON,NOPI,ALIGN(2)
_LIB\$KEY0\$	20	NOVEC,NOWRT, RD, EXE, SHR, LCL, REL, CON, PIC,ALIGN(1)
_LIB\$STATES	340	NOVEC,NOWRT, RD, EXE, SHR, LCL, REL, CON, PIC,ALIGN(1)
_LIB\$KEY1\$	48	NOVEC,NOWRT, RD, EXE, SHR, LCL, REL, CON, PIC,ALIGN(1)

## Library Statistics

File	----- Total	Symbols Loaded	----- Percent	Pages Mapped	Processing Time
------	----------------	-------------------	------------------	-----------------	--------------------

JBCCMDPRS  
V04-000

Job Controller Command Parsing Utilities

E 9  
16-Sep-1984 00:09:18  
14-Sep-1984 12:08:34

VAX-11 Bliss-32 V4.0-742  
[CLIUTL.SRC]JBCCMDPRS.B32;1

Page 109  
(45)

: \$255\$DUA28:[SYSLIB]LIB.L32;1	18619	115	0	1000	00:01.7
: \$255\$DUA28:[SYSLIB]TPAMAC.L32;1	42	25	59	14	00:00.1

COMMAND QUALIFIERS

: BLISS/CHECK=(FIELD,INITIAL,OPTIMIZE)/LIS=LIS\$:JBCCMDPRS/OBJ=OBJ\$:JBCCMDPRS MSRC\$:JBCCMDPRS/UPDATE=(ENH\$:JBCCMDPRS)

: Size: 3855 code + 430 data bytes  
: Run Time: 01:37.7  
: Elapsed Time: 05:00.5  
: Lines/CPU Min: 1885  
: Lexemes/CPU-Min: 40055  
: Memory Used: 245 pages  
: Compilation Complete



0049

AH-BT13A-SE  
VAX/VMS V4.0

**DIGITAL EQUIPMENT CORPORATION**  
**CONFIDENTIAL AND PROPRIETARY**



0050 AH-BT13A-SE  
VAX/VMS V4.0

DIGITAL EQUIPMENT CORPORATION  
CONFIDENTIAL AND PROPRIETARY

PRONOUNCE  
LIS

QUEMAN  
LIS

MATCHKEY  
LIS

PUTCLMSG  
LIS

QUEMANMSG  
LIS

PASSWORDS  
LIS

QUEMANSHO  
LIS